International Journal of Systems Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tsys20

A regularised fast recursive algorithm for fraction model identification of nonlinear dynamic systems

Li Zhang, Kang Li, Dajun Du, Yihuan Li & Minrui Fei

To cite this article: Li Zhang, Kang Li, Dajun Du, Yihuan Li & Minrui Fei (2023) A regularised fast recursive algorithm for fraction model identification of nonlinear dynamic systems, International Journal of Systems Science, 54:7, 1616-1638, DOI: 10.1080/00207721.2023.2188983

To link to this article: https://doi.org/10.1080/00207721.2023.2188983

Published online: 18 Mar 2023.

INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE
2023, VOL. 54, NO. 7, 1616–1638
https://doi.org/10.1080/00207721.2023.2188983

Taylor & Francis
Taylor & Francis Group

A regularised fast recursive algorithm for fraction model identification of nonlinear dynamic systems

Li Zhang[a], Kang Li[b], Dajun Du [a], Yihuan Li[c] and Minrui Fei[a]

[a]Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, People's Republic of China; [b]School of Electronic and Electrical Engineering, University of Leeds, Leeds, UK; [c]School of Control and Computer Engineering, North China Electric Power University, Beijing, People's Republic of China

ABSTRACT

The fraction model has been widely used to represent a range of engineering systems. To accurately identify the fraction model is however challenging, and this paper presents a regularised fast recursive algorithm (RFRA) to identify both the true fraction model structure and the associated unknown model parameters. This is achieved first by transforming the fraction form to a linear combination of nonlinear model terms. Then the terms in the denominator are used to form a regularisation term in the cost function to offset the bias induced by the linear transformation. According to the structural risk minimisation principle based on the new cost function, the model terms are selected based on their contributions to the cost function and the coefficients are then identified recursively without explicitly solving the inverse matrix. The proposed method is proved to have low computational complexity. Simulation results confirm the efficacy of the method in fast identification of the true fraction models for the targeted nonlinear systems.

ARTICLE HISTORY
Received 18 April 2022
Accepted 4 March 2023

KEYWORDS
Regularisation; fraction models; regularised fast recursive algorithm; nonlinear model identification

1. Introduction

Machine learning is a powerful tool for technological progress and has been widely adopted for approximating nonlinear dynamic systems (Yildiz et al., 2017). The nonlinear system modelling has been a subject of intensive research in the past decades, and various machine learning regression (Singh & Dhiman, 2021) models have been proposed, such as polynomial nonlinear models (Paduart et al., 2010; Wang et al., 2021; C. Yang et al., 2020), rational models (Austin et al., 2021; Zhu et al., 2015, 2019), support vector machine (SVM) based regression models (Liang & Zhang, 2021; Suykens et al., 2002) and neural network models (Du et al., 2022; Qiao et al., 2018; Z. Yang et al., 2020), to just name a few.

Among them, polynomial nonlinear regression models are the most popular machine learning regression methods due to their simplicity to develop and capture complex relationships between system inputs and system outputs (Gambella et al., 2021). In particular, the rational model which is a ratio of two polynomials has attracted more attentions in the community as many engineering systems can be modelled as the ratio of two linear-in-the-parameter equations (Andreev & Borulko, 2017; Curtain & Morris, 2009; Hussein & Batarseh, 2011; Kambhampati et al., 2000; Swarnakar et al., 2017, Dec; Zhu et al., 2019), each consisting of some base functions. They are often referred to fraction models, and the rational models which all consist of polynomial terms in both numerator and denominator is a special case (Li, 2005). The fraction model has the capability of concisely approximating a wide range of nonlinear systems, while traditional models such as nonlinear polynomial ARX models, neural networks, support vector machines (SVM) (J. Zhang et al., 2012) may become overly complex in order to achieve the same level of modelling accuracy.

To fit a fraction model using the training data is a machine learning regression modelling problem. To establish an accurate fraction model, an efficient automatic learning method is the key to both determine

CONTACT Kang Li k.li1@leeds.ac.uk School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, UK; Dajun Du ddj@i.shu.edu.cn Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, 200072, People's Republic of China

the model structure and identify unknown parameters. This is however technically challenging, and for rational models as a special case, the solutions fall into two major categories. The first one is to transform the rational form to a linear-in-the-parameter expression (Billings & Zhu, 1991, 1994). Based on the transformed system, various algorithms have been developed to identify a suitable model structure and in the meantime to identify the unknown linear coefficients (Billings & Mao, 1998; Billings & Zhu, 1991; Zhu, 2003, 2005; Zhu & Billings, 1996; Zhu et al., 2017). These algorithms all have the same aim to iteratively offset the bias by updating the coefficients and the regressors based on the variance of estimation errors. The resultant models identified by these well-established methods are often less complex, though they still suffer from high computation complexity. The other approach is directly selecting the base function using the orthogonal methods, such as an orthogonal rational model estimation (Billings & Zhu, 1994) and Stieltjes process (Austin et al., 2021).

In particular, the popular orthogonal least square (OLS) method has been widely used for identifying a range of nonlinear models including the rational models (Zhu & Billings, 1996) although the conventional OLS scheme is still computationally expensive, particularly for backward model reduction. To reduce the computational complexity, Li et al proposed a fast recursive algorithm (FRA) (Li et al., 2005). The FRA framework has further led to the development of a series of effective algorithms for nonlinear system modelling, such as the two-stage stepwise algorithm (Li et al., 2006), two-stage OLS method (L. Zhang et al., 2015), least squares support vector machine (LSSVM) (Zhao et al., 2015) as well as methods to construct compact RBF networks (Deng et al., 2012, 2013, 2010). This paper will use the similar algorithmic framework to build fraction models for nonlinear systems, aiming to both develop a compact model and reduce the computational complexity.

A challenge for identifying the fraction model is to eliminate the bias caused by the errors introduced by the transformation. To tackle this challenge, various methods have been proposed to estimate the errors (J. Chen et al., 2018; Jin & Weng, 2019; Manela & Biess, 2021; Piga & Tóth, 2014). In order to penalise the bias induced by the linear transformation, this paper adopts a negative regularisation term to penalise the errors introduced into the transformed linear model. In essence, the regularisation term is used to balance the linear model estimation error and the bias of the estimated parameters due to the linear transformation (Ohlsson et al., 2010). The regularisation method proposed by Tikhonov (Tikhonov & Glasko, 1965) and its variants (Luo et al., 2016; Münker & Nelles, 2016; L. Zhang et al., 2019) have been widely used in machine learning to reduce the bias introduced into the parameter estimation. Popular regularisation methods include the ℓ_0 regularisation, ℓ_1 regularisation and ℓ_2 regularisation (Castella et al., 2018; Luo et al., 2016; Senhaji et al., 2020; Tao et al., 2015; Torres-Barrán et al., 2018) where the weight of the regularisation term is often set positive to avoid over-fitting, while in this paper a negative weight is applied to the penalty term to reduce the bias introduced by the linear transformation. However, the weight has to be sufficiently small to ensure that the contributions to the cost function are positive.

Within the regularisation framework, to choose a suitable regularisation term is critical, and a number of kernel functions have been used to define the regularisation term (Birpoutsoukis et al., 2017; T. Chen et al., 2014; Pillonetto, 2018). The kernel method which performs a nonlinear data transformation has been widely applied in system identification, machine learning and function estimation (Bai et al., 2014; T. Chen et al., 2014; Hofmann et al., 2008; Pillonetto & De Nicolao, 2010; Pillonetto et al., 2014). Reproducing kernel Hilbert spaces (RKHSs) (Pillonetto et al., 2014; Valencia & lvarez, 2017) is a popular one. Similarly, in this paper, the inner product of nonlinear mapping terms in the denominator is added to the cost function with the aim to reduce the bias caused by the linear transformation. The effectiveness of this treatment will be assessed in the simulation section.

Given the quadratic cost function as a trade-off between the linear model error and model parameters bias caused by linear transformation, a regularised fast recursive algorithm is proposed in this paper to solve the optimisation problem. The main contributions in this paper are summarised as follows:

- A duality formulation is introduced to translate the fraction model identification problem to a quadratic linear regression problem. Based on the structural risk minimisation principle, the dual linear regression problem is minimised by a variant of the least squares solution. In essence, identification of a fraction model is thus translated to solving a linear regression problem, thus significantly reduced the computational complexity.

- To eliminate the bias induced by the linear-in-the-parameters transformation and to recover the true fraction model rather than identify a parsimonious model, a regularisation term is added to the cost function. The optimisation problem can be easily solved by a regularised least-squares method.
- To automatically identify both the fraction model coefficients and the fraction model structure, a regularised fast approach (RFRA) is proposed based on the original fast recursive algorithm (Li et al., 2005). Unlike the locally regularised automatic construction (LRAC) method (Du et al., 2012), the regularisation matrix does not have to be a diagonal matrix when using the novel regularised fast recursive method. In order to efficiently evaluate the candidate base function, a recursive net contribution is developed. The proposed RFRA method simplifies the calculation of the net contribution of model terms and computation of the least squares equations for the coefficients by defining a set of recursive matrices, and the fraction model structure is determined according to the net contribution easily.
- Finally, the complexity of the proposed method is analysed and compared with the well-known orthogonal method proposed in Zhu and Billings (1996) to confirm its efficiency.

The remainder of the paper is organised as follows. The fraction model identification problem is formulated in Section 2. Section 3 gives some necessary preliminaries for the proposed algorithm. Then a regularised fast method is detailed in Section 4, covering the principle of the new method, fast computation of the net contribution of model terms, fast computation of model coefficients, and computational complexity analysis. Simulation case studies are presented in Section 5. Finally, Section 6 concludes this paper.

2. Problem formulation

2.1. Fraction model

Considering a multiple-input single-output system, the fraction model can be expressed as the ratio of two linear-in-the-parameter models.

$$y(t) = \frac{a(t)}{c(t)} + e(t) = \frac{\sum_{i=1}^{r}[p_{ni}(\mathbf{x}(t)) \cdot \theta_{ni}]}{\sum_{j=1}^{h}[p_{dj}(\mathbf{x}(t)) \cdot \theta_{dj}]} + e(t) \tag{1}$$

where $y(t)$, $\mathbf{x}(t) = [x_1(t), \ldots, x_q(t)]$ and $e(t)$ are the model output, the model input vector with q dimension and the model error at time instant t, respectively. p_{ni} represents the ith nonlinear base mapping term of the numerator and p_{dj} represents the jth nonlinear base mapping term of the denominator. r and h denote the numbers of the nonlinear terms in the numerator and in the denominator of the fraction model respectively. Also θ_{ni} and θ_{dj} represent the coefficients of the ith regressor in the numerator and the jth regressor in the denominator respectively.

To identify the fraction model from the training data, the system (1) can be expressed in the matrix representation first. Then the cost function of the corresponding dual quadratic linear regression problem can be calculated and further simplified. Suppose N data samples $\{x(t), y(t)\}_{t=1}^{N}$ are used for model identification, Equation (1) can be formulated as

$$\boldsymbol{y} = \boldsymbol{a} \oslash \boldsymbol{c} + \boldsymbol{e} = (\boldsymbol{P}_n\boldsymbol{\Theta}_n) \oslash (\boldsymbol{P}_d\boldsymbol{\Theta}_d) + \boldsymbol{e} \tag{2}$$

where $\boldsymbol{y} = [y(1), \ldots, y(N)]^{\mathrm{T}}$ is the model output vector. $\boldsymbol{a} = [a(1), \ldots, a(N)]^{\mathrm{T}}$ is the numerator sample vector. $\boldsymbol{c} = [c(1), \ldots, c(N)]^{\mathrm{T}}$ is the denominator sample vector. The components of $\boldsymbol{a} \oslash \boldsymbol{c}$ are $(\boldsymbol{a} \oslash \boldsymbol{c})_l = a(l)/c(l)$, $1 =< l <= N$. $\boldsymbol{e} = [e(1), \ldots, e(N)]^{(\mathrm{T})}$ is the model error vector. $\boldsymbol{P}_n = [\boldsymbol{p}_{n1}, \ldots, \boldsymbol{p}_{nr}] \in \Re^{N\times r}$ is the selected nonlinear mapping matrix for the numerator with $\boldsymbol{p}_{ni} = [p_{ni}(\boldsymbol{x}(1)), \ldots p_{ni}(\boldsymbol{x}(N))]^{\mathrm{T}}, i = 1, \ldots, r$. $\boldsymbol{\Theta}_n = [\theta_{n1}, \ldots, \theta_{nr}]^{\mathrm{T}} \in \Re^{r\times 1}$ is the coefficients vector for the numerator. $\boldsymbol{P}_d = [\boldsymbol{p}_{d1}, \ldots, \boldsymbol{p}_{dh}] \in \Re^{N\times h}$ is the selected nonlinear mapping matrix for the denominator with $\boldsymbol{p}_{dj} = [p_{dj}(\boldsymbol{x}(1)), \ldots, p_{dj}(\boldsymbol{x}(N))]^{\mathrm{T}}$, $j = 1, \ldots, h$. $\boldsymbol{\Theta}_d = [\theta_{d1}, \ldots, \theta_{dh}]^{\mathrm{T}} \in \Re^{h\times 1}$ is the coefficients vector for the denominator.

2.2. Transformation to linear-in-the-parameter formulation

Fraction models are not naturally linear in the parameters and its identification presents a challenge. To identify fraction models, an elementwise product of the two vectors is first defined, namely Hadamard product using $\odot$ to denote the product (Nielsen, 2019).

Suppose $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two vectors of the same dimension, then the lth component of $\boldsymbol{\alpha} \odot \boldsymbol{\beta}$ is $(\boldsymbol{\alpha} \odot \boldsymbol{\beta})_l = \alpha(l)\beta(l)$.

To obtain the matrix representation of the corresponding linear system in the sense of least squares, both sides of Equation (2) are Hadamard multiplied by '$\boldsymbol{c} \in \Re^{N\times 1}$', thus:

$$\begin{aligned}\boldsymbol{y} \odot \boldsymbol{c} &= \boldsymbol{a} \oslash \boldsymbol{c} \odot \boldsymbol{c} + \boldsymbol{e} \odot \boldsymbol{c} = \boldsymbol{a} + \boldsymbol{e} \odot \boldsymbol{c}\\ &= (\boldsymbol{P}_n\boldsymbol{\Theta}_n) + \boldsymbol{e} \odot \boldsymbol{c} = \boldsymbol{y} \odot (\boldsymbol{P}_d\boldsymbol{\Theta}_d) \qquad (3)\end{aligned}$$

Expanding $\boldsymbol{P}_d\boldsymbol{\Theta}_d$ and $\boldsymbol{P}_n\boldsymbol{\Theta}_n$ in (3) gives

$$\begin{aligned}&\boldsymbol{y} \odot (\boldsymbol{p}_{d1} \cdot \theta_{d1} + \boldsymbol{p}_{d2} \cdot \theta_{d2} + \cdots + \boldsymbol{p}_{dh} \cdot \theta_{dh})\\ &\quad = (\boldsymbol{p}_{n1} \cdot \theta_{n1} + \boldsymbol{p}_{n2} \cdot \theta_{n2} + \cdots + \boldsymbol{p}_{nr} \cdot \theta_{nr}) + \boldsymbol{e} \odot \boldsymbol{c} \qquad (4)\end{aligned}$$

Thereby,

$$\begin{aligned}&\boldsymbol{y} \odot \boldsymbol{p}_{d1} \cdot \theta_{d1} + \boldsymbol{y} \odot \boldsymbol{p}_{d2} \cdot \theta_{d2} + \cdots + \boldsymbol{y} \odot \boldsymbol{p}_{dh} \cdot \theta_{dh}\\ &\quad = (\boldsymbol{p}_{n1} \cdot \theta_{n1} + \boldsymbol{p}_{n2} \cdot \theta_{n2} + \cdots + \boldsymbol{p}_{nr} \cdot \theta_{nr}) + \boldsymbol{e} \odot \boldsymbol{c} \qquad (5)\end{aligned}$$

Then all the terms except '$\boldsymbol{y} \odot \boldsymbol{p}_{d1} \cdot \theta_{d1}$' are moved to the right side, resulting in a linear-in-the-parameter model

$$\begin{aligned}\boldsymbol{y} \odot \boldsymbol{p}_{d1}\theta_{d1} &= (\boldsymbol{p}_{n1} \cdot \theta_{n1} + \boldsymbol{p}_{n2} \cdot \theta_{n2} + \cdots + \boldsymbol{p}_{nr} \cdot \theta_{nr})\\ &\quad - \boldsymbol{y} \odot \boldsymbol{p}_{d2} \cdot \theta_{d2} - \cdots - \boldsymbol{y} \odot \boldsymbol{p}_{dh} \cdot \theta_{dh}\\ &= [\boldsymbol{p}_{n1}, \boldsymbol{p}_{n2}, \ldots, \boldsymbol{p}_{nr} \cdot \theta_{nr}]\\ &\quad \times [\theta_{n1}, \theta_{n2}, \ldots, \theta_{nr}]^{\mathrm{T}}\\ &\quad - [\boldsymbol{y} \odot \boldsymbol{p}_{d2}, \cdots, \boldsymbol{y} \odot \boldsymbol{p}_{dh}]\\ &\quad \cdot [\theta_{d2}, \ldots, \theta_{dh}]^{\mathrm{T}} + \boldsymbol{e} \odot \boldsymbol{c}\\ &= [\boldsymbol{p}_{n1}, \ldots, \boldsymbol{p}_{nr}, -\boldsymbol{y} \odot \boldsymbol{p}_{d2}, \ldots, -\boldsymbol{y} \odot \boldsymbol{p}_{dh}]\\ &\quad \times [\theta_{n1}, \ldots, \theta_{nr}, \theta_{d2}, \ldots, \theta_{dh}]^{\mathrm{T}}\\ &\quad + \boldsymbol{e} \odot \boldsymbol{c} \qquad (6)\end{aligned}$$

By defining $\boldsymbol{Y} = \boldsymbol{y} \odot \boldsymbol{p}_{d1}\theta_{d1}$, $\boldsymbol{\Phi} = [\boldsymbol{p}_{n1}, \ldots, \boldsymbol{p}_{nr}, -\boldsymbol{y} \odot \boldsymbol{p}_{d2}, \ldots, -\boldsymbol{y} \odot \boldsymbol{p}_{dh}]$ and $\boldsymbol{\Theta} = [\theta_{n1}, \ldots, \theta_{nr}, \theta_{d2}, \ldots, \theta_{dh}]^{\mathrm{T}}$, the system can be reformulated as follows:

$$\boldsymbol{Y} = \boldsymbol{\Phi\Theta} + \boldsymbol{c} \odot \boldsymbol{e} = \boldsymbol{\Phi\Theta} + \boldsymbol{\Xi} \qquad (7)$$

where θ_{d1} generally takes the value of 1, $\boldsymbol{p}_{d1} = \mathbf{1}_{N\times 1}$, $\boldsymbol{Y} = [y(1) \cdot p_{d1}(1), \ldots, y(N) \cdot p_{d1}(N)]^{\mathrm{T}} = [y(1), \ldots, y(N)]^{\mathrm{T}}$ is the output vector of new system, $\boldsymbol{\Theta} = [\theta_{n1}, \cdots, \theta_{nr}, \theta_{d2}, \ldots, \theta_{dh}]^{\mathrm{T}}$ is the new system linear parameter vector, $\boldsymbol{\Xi} = [c(1)e(1), \ldots, c(N)e(N)]^{\mathrm{T}}$ is the coloured noise of the new system, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M] = [\boldsymbol{p}_{n1}, \ldots, \boldsymbol{p}_{nr}, -\boldsymbol{y} \odot \boldsymbol{p}_{d2}, \ldots, -\boldsymbol{y} \odot \boldsymbol{p}_{dh}]$ with M regressors for inclusion in the model, where $M = r + h - 1$.

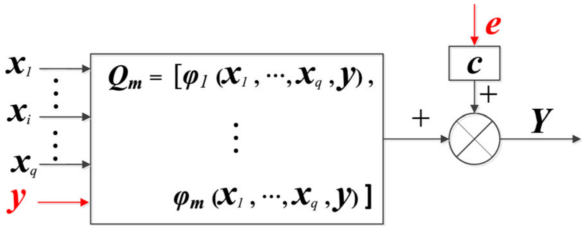
$\boldsymbol{x}_1$
$\boldsymbol{x}_i$
$\boldsymbol{x}_q$
$\boldsymbol{y}$
$\boldsymbol{Q}_m = [\varphi_1(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_q, \boldsymbol{y}),$
$\vdots$
$\varphi_m(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_q, \boldsymbol{y})]$
$\boldsymbol{e}$
$\boldsymbol{c}$
+
+
$\boldsymbol{Y}$

Figure 1. The structure of the new linear-in-parameters system.

2.3. Model identification challenges

By transforming the fraction model in the linear-in-the-parameter form, the model identification is simplified to a linear regression problem. Both the least squares method and the likelihood estimation are two well-established methods. Here the least squares method (LS) is applied to model equation (7) given that the distribution of the training samples is unknown prior.

For the new system formulated in Equation (7), the input matrix and output vector of the model are $[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q, \boldsymbol{y}] \in \Re^{N\times(q+1)}$ and $\boldsymbol{Y} \in \Re^{N\times 1}$, respectively. Then m candidate nonlinear features $\boldsymbol{\varphi}_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q, \boldsymbol{y})$, $i = 1, \ldots, m$ are generated if the true system is unknown a priori. Generally speaking, if the true model is unknown, then various parsimonious nonlinear models such as polynomial models and other linear-in-the-parameter models can be built to achieve equivalent modelling performance under certain conditions (Li & Peng, 2007). Thus, a polynomial is often used to approximate a range of nonlinear systems. To this end, $\boldsymbol{\varphi}_i$ can be defined as a polynomial term. If the system model is partially known a priori, then fundamental knowledge can be obtained for an engineering system (Li, 2005) based on which fundamental functions ($\boldsymbol{\varphi}_i$) can be identified, such as exponential, fraction order polynomial terms, trigonometric, or rational functions, etc. The resultant model structure is illustrated in Figure 1.

In statistics, there exists biased parameter estimators to dominate any unbiased parameter estimator in terms of the mean squared error (MSE) (Hong et al., 2008), therefore there is always a requirement to balance the variance and bias using the least square method. In LS method, the linear coefficients $\mathbf{\Theta}$ can be estimated by

$$\hat{\mathbf{\Theta}} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\boldsymbol{Y} \tag{8}$$

In theory, for the linear-in-the-parameter system subject to white noise, the least square solution of the coefficients ($E[\hat{\mathbf{\Theta}}]$) is equal to the true value $\mathbf{\Theta}$ if the sample number is sufficiently large. However, for the system in Equation (7), the linear coefficients in Equation (8) can be described as

$$\hat{\mathbf{\Theta}} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}(\mathbf{\Phi}\mathbf{\Theta} + \Xi) = \mathbf{\Theta} + (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\Xi \tag{9}$$

It is clear that the noise Ξ in Equation (9) comes from $\boldsymbol{c} \odot \boldsymbol{e}$ defined in Equation (7), therefore it is a coloured noise. From Figure 1, $\mathbf{\Phi}$ is correlated with $\boldsymbol{e}$ as it is included in the information of the output $\boldsymbol{y}$. Therefore, the expectation of estimation coefficients ($E[\hat{\mathbf{\Theta}}]$) is $\mathbf{\Theta} + E[(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\Xi]$ rather than $\mathbf{\Theta}$. Thus, the least square method applied to system (7) has a bias term in the solution, and it is necessary to eliminate the bias to improve the generalisation performance of a parsimonious model or to identify the true system model from a model candidate term pool consisting of the true model terms.

From Equation (9), it is clear that the bias is related to the selected terms of the denominator and $\boldsymbol{e}$. Accordingly, an improved method is proposed in the following to eliminate the bias.

3. Preliminaries

Through the linear transformation, noises have been introduced into both input and outputs of the new system (2). Furthermore, the output-error is a coloured noise which is the response excited by the white noise of the original system. Given the coloured noise, the conventional recursive least square method leads to a biased solution. To achieve unbiased coefficients estimation, a similar process to the reproducing kernel Hilbert space (RKHS) method (Pillonetto, 2018) which uses the inner products of the denominator is employed to mitigate the contribution of the bias to the cost function. By adding a regularisation term with a negative weight, it is possible to eliminate the bias introduced into the estimation of the original system.

3.1. The original cost function

Lemma 3.1: *Suppose $\boldsymbol{e} = [e(1), \ldots, e(N)]$ obeys a Gaussian distribution with zero mean, for given sufficient enough data samples, then Hadamard product $((\boldsymbol{c} \odot \boldsymbol{e})^{\mathrm{T}}(\boldsymbol{c} \odot \boldsymbol{e}))$ satisfies:*

$$(\boldsymbol{c} \odot \boldsymbol{e})^{\mathrm{T}}(\boldsymbol{c} \odot \boldsymbol{e}) = \Xi^{\mathrm{T}}\Xi = \frac{1}{N} \cdot \boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \cdot \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} \tag{10}$$

The proof is given as follows.

According to the multiplication of matrices, the right side of Equation (10) can be expanded as

$$\begin{aligned}
\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \cdot \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} &= \sum_{i=1}^{N} c^2(i) \sum_{j=1}^{N} \mathrm{e}^2(i) \\
&= (c^2(1) + \cdots + c^2(N)) \sum_{j=1}^{N} \mathrm{e}^2(j) \\
&= c^2(1) \sum_{j=1}^{N} \mathrm{e}^2(j) + \cdots + c^2(N) \sum_{j=1}^{N} \mathrm{e}^2(j) \\
&= (\boldsymbol{c} \odot \boldsymbol{e})^{\mathrm{T}}(\boldsymbol{c} \odot \boldsymbol{e}) + c^2(1) \sum_{j=2}^{N} \mathrm{e}^2(j) + \cdots \\
&\quad + c^2(i) \sum_{j=1\&j\neq i}^{N} \mathrm{e}^2(j) + \cdots \\
&\quad + c^2(N) \sum_{j=1}^{N-1} \mathrm{e}^2(j)
\end{aligned} \tag{11}$$

If the system error '$\boldsymbol{e}$' is a white noise with the variance 'σ_e^2', Equation (11) can be reformulated as follows given that there are sufficient data samples.

$$\begin{aligned}
\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \cdot \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} &= \Xi^{\mathrm{T}}\Xi + c^2(1) \cdot (N-1) \cdot \sigma_e^2 + \cdots \\
&\quad + c^2(i) \cdot (N-1) \cdot \sigma_e^2 + \cdots \\
&\quad + c^2(N) \cdot (N-1) \cdot \sigma_e^2 \\
&= \Xi^{\mathrm{T}}\Xi + \sum_{i=1}^{N} c^2(i) \cdot (N-1) \cdot \sigma_e^2 \\
&= \Xi^{\mathrm{T}}\Xi + \sum_{i=1}^{N} c^2(i) \cdot \frac{N-1}{N}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} \\
&= \Xi^{\mathrm{T}}\Xi + \frac{N-1}{N}\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \cdot \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}
\end{aligned} \tag{12}$$

Accordingly, Equation (10) is true.

Considering the coloured noise ’Ξ’ in Equation (7), the sum of squared coloured noises can be calculated as follows:

$$\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} = \frac{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}\cdot\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}}{N} \tag{13}$$

where $\boldsymbol{\Xi} = \boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\Theta}$ and $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{a} \oslash \boldsymbol{c}$ are the linear-in-the-parameter system error vector and the original system error vector respectively. To estimate coefficients ($\boldsymbol{\Theta}$), the denominator ($\boldsymbol{c}$) of the original system in Equation (2) can be written as follows:

$$\boldsymbol{c} = \boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E} \tag{14}$$

where $\boldsymbol{\Psi}_{N\times M}$ with $M = r + h - 1$ and $\boldsymbol{E}$ together represent the nonlinear mapping matrix $\boldsymbol{P}_d$ in Equation (2). M represents the number of the regressors for inclusion in the model. $\boldsymbol{E} = \boldsymbol{1}_{N\times 1}$ with size $N \times 1$ represent the constant bias ($\boldsymbol{p}_{d1}$) of the denominator on the left. $\boldsymbol{\Psi}_{N\times M} = [\boldsymbol{\psi}_1 \cdots \boldsymbol{\psi}_M] = [\boldsymbol{0} \cdots \boldsymbol{0}\ \boldsymbol{p}_{d2} \cdots \boldsymbol{p}_{dh}]$ is augmented from the terms ($[\boldsymbol{p}_{d2}, \ldots, \boldsymbol{p}_{dh}]$) of the denominator being moved to the right side, hence they are referred to augmented terms in the following context. And

$$\boldsymbol{\psi}_\rho = \begin{cases} 0 & \rho = 1, \ldots, r \\ \boldsymbol{p}_{dj} & \rho = r+1, \ldots, M; \quad j = 2, \ldots, h \end{cases}$$

Notice from Equation (13) that the cost function using the sum of the squared fraction system errors can be computed by:

$$\boldsymbol{J}_o = \boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} = \frac{N\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi}}{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}} \tag{15}$$

3.2. The dual solution

To minimise the modelling error, Equation (15) is often used as the criterion. However, the cost function is nonlinear with respect to the model coefficients. To this end, a linear optimisation algorithm is a well-known approach for many nonlinear problems, for example, the least squares SVM is identified by solving linear equations in the dual instead of solving a quadratic programming problem with inequality constraints in the primal (Mall & Suykens, 2015). Thus, instead of using a complex nonlinear derivative-based methods, similar to Pillonetto et al. (2014), a linear dual problem combining the original linear cost function and the inner product of the denominator sample vector, is proposed as the cost function which is expressed in Proposition 3.1.

Proposition 3.1: *Subject to the existence of ’λ’ which satisfies* ’$\min(\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}) = 0$’, *minimising the cost function* (16) *for the transformed model* (7) *is equivalent to minimising* (15) *for the original model* (2).

$$\begin{aligned}\boldsymbol{J} &= \boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c} \\ &= (\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\Theta})^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\Theta}) \\ &\quad + \lambda(\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E}) \\ &= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta} + \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta} \\ &\quad + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta} + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\end{aligned} \tag{16}$$

where $\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi}$ is used to evaluate the estimation error of the transformed linear-in-the-parameter system in Equation (7). From the expansion of Equation (16), $\lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta} + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta}$ includes the regularisation parameter λ and the ℓ_1 and ℓ_2 norms of the estimated coefficients ($\boldsymbol{\Theta}$). Thus, $\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}$ with the regularisation parameter λ is used as the regularisation penalty term. Meanwhile, ’$\min(\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}) = 0$’ is the minimum value of the cost function (16), i.e. the optimal cost function.

The proof of the Proposition 3.1 is given below.

Considering the new cost function, the minimum estimation is given when the partial derivative of the cost function with respect to the coefficients is equal to zero. Thus, the coefficients can be estimated using:

$$\frac{\partial\boldsymbol{J}}{\partial\boldsymbol{\Theta}} = -2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} + 2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} \tag{17}$$

Setting Equation (17) equal to zero, the optimal coefficients are given as

$$\hat{\boldsymbol{\Theta}} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}) \tag{18}$$

Similarly, for the nonlinear original cost function in Equation (15), its partial derivative with respect to the coefficients is calculated as

$$\begin{aligned}\frac{\partial\boldsymbol{J}_o}{\partial\boldsymbol{\Theta}} &= \frac{\partial((\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\Theta})^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{\Theta})/(\boldsymbol{\Psi}\boldsymbol{\Theta})^{\mathrm{T}}(\boldsymbol{\Psi}\boldsymbol{\Theta}))}{\partial\boldsymbol{\Theta}} \\ &= N\left(\frac{\begin{array}{c}(-2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} + 2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta}) \\ ((\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E}))\end{array}}{((\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\boldsymbol{\Theta} + \boldsymbol{E}))^2}\right.\end{aligned}$$

$$\left. - \frac{((\boldsymbol{Y}-\boldsymbol{\Phi\Theta})^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{\Phi\Theta}))(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))}{((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))^2} \right) \tag{19}$$

To calculate the partial derivative in Equation (19) using the optimal coefficients, $2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}$ is added in the first part, thus

$$\begin{aligned}\frac{\partial \boldsymbol{J}_o}{\partial \boldsymbol{\Theta}} = N \Bigg(& \frac{(-2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} + 2\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))}{((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))^2} \\ & - \frac{((\boldsymbol{Y}-\boldsymbol{\Phi\Theta})^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{\Phi\Theta}))(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))}{((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))^2} \\ & - \frac{(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi\Theta} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))}{((\boldsymbol{\Psi\Theta}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi\Theta}+\boldsymbol{E}))^2} \Bigg)\end{aligned} \tag{20}$$

Obviously, the first part of the numerator in Equation (20) is the partial derivative in Equation (17). Substituting Equation (18) into Equation (20), the first part of Equation (20) is zero and it can be rewritten as follows after N is left out:

$$\begin{aligned}\frac{\partial \boldsymbol{J}_o}{\partial \boldsymbol{\Theta}} = & -\frac{((\boldsymbol{Y}-\boldsymbol{\Phi}\hat{\boldsymbol{\Theta}})^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{\Phi}\hat{\boldsymbol{\Theta}}))(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))}{((\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))^2} \\ & - \frac{(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}} + 2\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})((\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))}{((\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))^2}\end{aligned} \tag{21}$$

Since both $(\boldsymbol{Y}-\boldsymbol{\Phi}\hat{\boldsymbol{\Theta}})^{\mathrm{T}}(\boldsymbol{Y}-\boldsymbol{\Phi}\hat{\boldsymbol{\Theta}})$ and $(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}$ $(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})$ are scalars, extracting $2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})$ after changing its position, then (21) can be expressed as

$$\begin{aligned}\frac{\partial \boldsymbol{J}_o}{\partial \boldsymbol{\Theta}} & = -\frac{(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))(\min(\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}))}{((\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))^2} \\ & = -\frac{(2\lambda\boldsymbol{\Psi}^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))(\min(\boldsymbol{J}))}{((\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{\Psi}\hat{\boldsymbol{\Theta}}+\boldsymbol{E}))^2}\end{aligned} \tag{22}$$

Now setting Equation (22) equal to zero, yielding

$$\min(\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}) = 0 \tag{23}$$

The above derivations show that if there exists a negative number λ which makes the dual problem in Equation (16) of system (7) equal to zero for the optimal coefficients, then the minimum of $\boldsymbol{J}$ is zero, and Equations (22) and (23) are satisfied. Thus, λ should be optimised to ensure that its absolute value is sufficiently small such that $(\boldsymbol{\Xi}^{\mathrm{T}}\boldsymbol{\Xi} + \lambda\boldsymbol{c}^{\mathrm{T}}\boldsymbol{c}) >= 0$, and our study shows that the order of its magnitude is generally no greater than -4. This shows that the identification of the fraction system is equivalent to optimising the proposed dual problem, and the proposed dual problem (16) can be solved based on the regularised least squares. It should be noted that the regularisation penalty term $(\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})$ has only three parts and they are only related to part of the model coefficients. This is different from conventional regularisation problem where penalty is applied to all model coefficients. To reduce the computation complexity of the regularised least squares, this paper proposes to simplify the general regularised least squares according to the FRA framework.

4. Regularised fast recursive algorithm

According to Section 3, the fraction system is identifiable using the proposed primal–dual solution. To solve the transformed quadratic cost function, the least squares methods are most popular. In particular, the orthogonal least squares (OLS) (Zhu & Billings, 1996) and the fast recursive algorithm (FRA) (Li et al., 2006, 2005) can not only be used for parameters estimation, but also for model structure selection. Among them, the FRA method solves the least squares problem recursively without requiring matrix decomposition, and it further allows both fast forward and backward model identification. In the following, an regularised fast recursive method is presented.

4.1. The principle of regularised fast recursive method

4.1.1. Model term selection based on the regularised cost function

Referring to Li et al. (2005), a fast calculation of the net contribution of each term chosen into the model is a key. As discussed earlier, the cost function can be defined as

$$\begin{aligned}\boldsymbol{J} = & \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi\Theta} + \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi\Theta} \\ & + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi\Theta} + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi\Theta} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\end{aligned} \tag{24}$$

Substituting the optimal coefficients Equation (18) into Equation (24), the corresponding minimal cost function is given as follows and the simplification procedure is given in Appendix 1:

$$\begin{aligned} J = \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}})\boldsymbol{Y} \\ + 2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{E} \\ + \lambda\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{I} - \lambda\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}})\boldsymbol{E} \end{aligned} \tag{25}$$

The elegance of the FRA approach lies in the efficient and stable calculation of the net decrease in the cost function when a candidate term is added to the model and the model coefficients can be efficiently obtained using backward substitution (Li et al., 2005). This framework also allows for both forward and backward model identification (Li et al., 2006). In this paper, to select the regressors and estimate the coefficients efficiently using RFRA, a proper selection criteria with the associated coefficient estimation method needs to be derived and implemented using a simplified and efficient computational framework.

Therefore, in the following, the net contribution of each model term will be derived first based on the cost function (25) for the RFRA. Then, the four stages of the RFRA method are introduced in detail. Firstly, a cost function is designed for the fraction system identification based on the dual transformation. Secondly, according to the conventional regularised least squares method, the selection criteria and the coefficients are calculated based on Equations (25) and (18). Thirdly, recursive matrices are defined to derive a simplified and efficient computational framework for calculating the net contributions of model terms and the model coefficients. Finally, to identify the fraction model using RFRA, the elements of recursive matrices can be updated recursively without incurring much calculations. Substituting these recursive matrices into the target equations, the regressors are selected and the coefficients are estimated respectively. The schematic of the proposed RFRA is illustrated in Figure 2.

Suppose the model (2) has a maximum of $(m+1)/2$ regression terms both in the numerator and the denominator, the matrix of candidate regression terms for (2) can be initially set as $\boldsymbol{\Omega}_{(m+1)/2} = [\mathbf{1}, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{(m-1)/2}]$. Then the transformed linear model (7) has a maximum of m regression terms because the constant term in the denominator should have been moved as a part of the output. The matrix of candidate regression terms in (7) can be initially set as $\boldsymbol{Q}_m = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_m] = [\mathbf{1}, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{(m-1)/2}, -\boldsymbol{y} \odot \boldsymbol{\omega}_1, \ldots, -\boldsymbol{y} \odot \boldsymbol{\omega}_{(m-1)/2}]$ and the corresponding auxiliary matrix for regularisation can be initially set as $\boldsymbol{\Upsilon}_m = [\boldsymbol{\kappa}_1, \ldots, \boldsymbol{\kappa}_m] = [\mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}, \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{(m-1)/2}]$ respectively, where the indices of the elements in $\boldsymbol{Q}_{m-k+1}$ and $\boldsymbol{\Upsilon}_{m-k+1}$ at the kth iteration start from k, and thus $\boldsymbol{\varphi}_i$ and $\boldsymbol{\kappa}_i$ are the $(i-k+1)$th candidate terms in the candidate regression matrix $\boldsymbol{Q}_{m-k+1}$ and the augmented matrix $\boldsymbol{\Upsilon}_{m-k+1}$, respectively.

After the kth regressors has been selected into the model (7), the selected regression term matrix is recorded as $\boldsymbol{\Phi}_k = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k]$ and the selected augmented term matrix is recorded as $\boldsymbol{\Psi}_k = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_k]$. Then the remaining candidate regressor matrix and the candidate augmented matrix are $\boldsymbol{Q}_{m-k} = [\boldsymbol{\varphi}_{k+1}, \ldots, \boldsymbol{\varphi}_m]$ and $\boldsymbol{\Upsilon}_{m-k} = [\boldsymbol{\kappa}_{k+1}, \ldots, \boldsymbol{\kappa}_m]$ respectively. Thus the optimal model coefficients and the corresponding cost function are given as

$$\hat{\boldsymbol{\Theta}}_k = (\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k)^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{E}) \tag{26}$$

$$\begin{aligned} J_k = \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{\Phi}_k(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k)^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}})\boldsymbol{Y} \\ + 2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Psi}_k(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k)^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{E} \\ + \lambda\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{I} - \lambda\boldsymbol{\Psi}_k(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k)^{-1}\boldsymbol{\Psi}_k^{\mathrm{T}})\boldsymbol{E} \end{aligned} \tag{27}$$

Before introducing the regularised fast recursive algorithm, a recursive matrix at the kth iteration is defined as

$$\boldsymbol{M}_k \triangleq \boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k \tag{28}$$

And the inversion of $\boldsymbol{M}_k^{-1} = (\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_k)^{-1}$ can be computed as follows

$$\begin{aligned} \boldsymbol{M}_k^{-1} &= \begin{bmatrix} \boldsymbol{M}_{k-1} & \begin{array}{c} \boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k \\ +\lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k \end{array} \\ \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_{k-1} + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_{k-1} & \boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \boldsymbol{F}_{k-1} & \boldsymbol{g}_{k-1} \\ \boldsymbol{g}_{k-1}^{\mathrm{T}} & f_{k-1} \end{bmatrix} \end{aligned} \tag{29}$$

where $\boldsymbol{\Phi}_{k-1} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{k-1}]$ and $\boldsymbol{\Psi}_{k-1} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{k-1}]$.

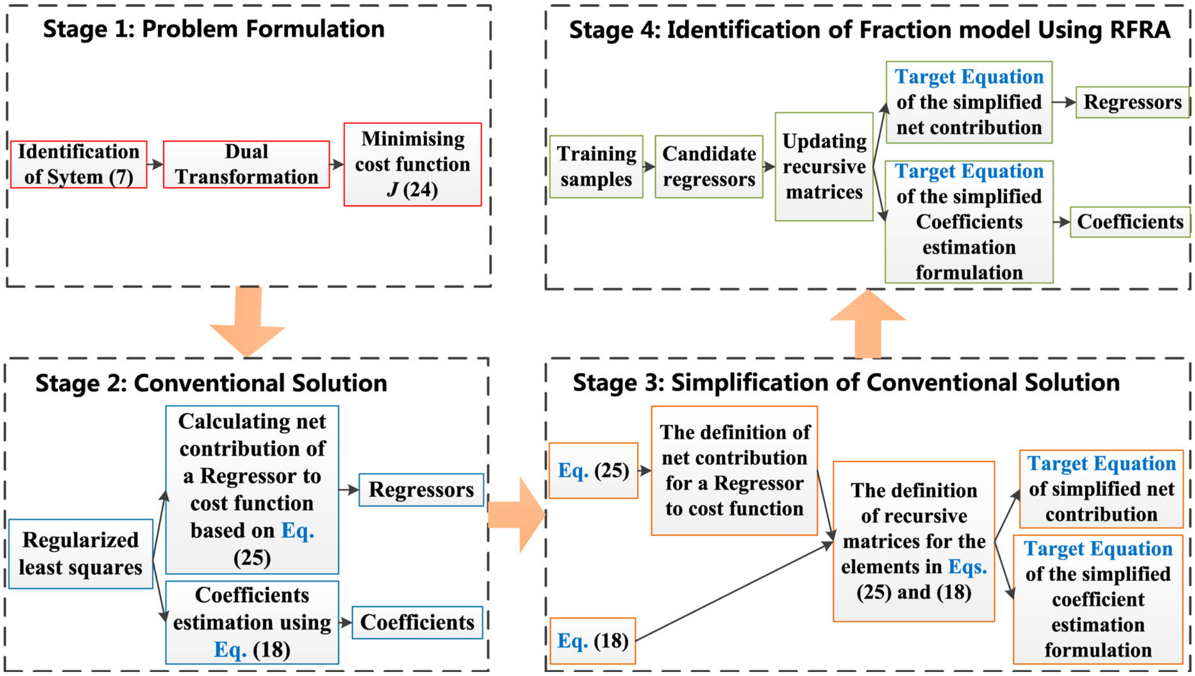
Stage 1: Problem Formulation
Identification of Sytem (7)
Dual Transformation
Minimising cost function J (24)
Stage 2: Conventional Solution
Regularized least squares
Calculating net contribution of a Regressor to cost function based on Eq. (25)
Regressors
Coefficients estimation using Eq. (18)
Coefficients
Stage 3: Simplification of Conventional Solution
Eq. (25)
The definition of net contribution for a Regressor to cost function
Eq. (18)
The definition of recursive matrices for the elements in Eqs. (25) and (18)
Target Equation of simplified net contribution
Target Equation of the simplified coefficient estimation formulation
Stage 4: Identification of Fraction model Using RFRA
Training samples
Candidate regressors
Updating recursive matrices
Target Equation of the simplified net contribution
Regressors
Target Equation of the simplified Coefficients estimation formulation
Coefficients

Figure 2. The schematic of the proposed RFRA method.

Given that $\boldsymbol{M}_k \cdot \boldsymbol{M}_k^{-1} = \boldsymbol{I}$, then $\boldsymbol{F}_{k-1}, \boldsymbol{g}_{k-1}$ and $\boldsymbol{f}_{k-1}$ can be calculated as follows

$$\begin{cases} \boldsymbol{F}_{k-1} = [\boldsymbol{M}_{k-1} - (\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k) \\ (\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_{k-1} + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_{k-1})/(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k)]^{-1} \\ \boldsymbol{g}_{k-1} = -\boldsymbol{F}_{k-1}(\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k) \\ /(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k) \\ \boldsymbol{f}_{k-1} = \dfrac{\begin{gathered}(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k) \\ +(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_{k-1} + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_{k-1})\boldsymbol{F}_{k-1} \\ (\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k)\end{gathered}}{(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k)^2} \end{cases} \tag{30}$$

According to the matrix inversion lemma $[\boldsymbol{A} + \boldsymbol{BCD}]^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}[\boldsymbol{DA}^{-1}\boldsymbol{B} + \boldsymbol{C}^{-1}]^{-1}\boldsymbol{DA}^{-1}$, $\boldsymbol{F}_{k-1}$ can be computed as follows

$$\boldsymbol{F}_{k-1} = \boldsymbol{M}_{k-1}^{-1} + \frac{\begin{gathered}\boldsymbol{M}_{k-1}^{-1}(\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k) \\ (\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_{k-1} + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_{k-1})\boldsymbol{M}_{k-1}^{-1}\end{gathered}}{\begin{gathered}\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\psi}_k \\ -(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\Phi}_{k-1} + \lambda\boldsymbol{\psi}_k^{\mathrm{T}}\boldsymbol{\Psi}_{k-1})\boldsymbol{M}_{k-1}^{-1} \\ (\boldsymbol{\Phi}_{k-1}^{\mathrm{T}}\boldsymbol{\phi}_k + \lambda\boldsymbol{\Psi}_{k-1}^{\mathrm{T}}\boldsymbol{\psi}_k)\end{gathered}} \tag{31}$$

Thus, after kth selection, the cost function of Equation (27) becomes

$$\begin{aligned} J_k &= \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}})\boldsymbol{Y} + 2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{E} \\ &\quad + \lambda\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{I} - \lambda\boldsymbol{\Psi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Psi}_k^{\mathrm{T}})\boldsymbol{E} \end{aligned} \tag{32}$$

Assuming the regression term $\boldsymbol{\varphi}_i$ is selected from the candidate matrix $\boldsymbol{Q}_{m-k}$ at $(k+1)$th selection, the selected regression term matrix becomes $\boldsymbol{\Phi}_{k+1} = [\boldsymbol{\Phi}_k, \boldsymbol{\varphi}_i]$ and the selected augmented regression term matrix is $\boldsymbol{\Psi}_{k+1} = [\boldsymbol{\Psi}_k, \boldsymbol{\kappa}_i]$, while the remaining candidate regressor matrix and the candidate augmented matrix are $\boldsymbol{Q}_{m-k-1} = [\boldsymbol{\varphi}_{k+1}, \ldots, \boldsymbol{\varphi}_{i-1}, \boldsymbol{\varphi}_{i+1}, \ldots, \boldsymbol{\varphi}_m]$ and $\boldsymbol{\Upsilon}_{m-k-1} = [\boldsymbol{\kappa}_{k+1}, \ldots, \boldsymbol{\kappa}_{i-1}, \boldsymbol{\kappa}_{i+1}, \ldots, \boldsymbol{\kappa}_m]$ respectively.

Thus, the cost function is updated as follows

$$\begin{aligned} J_{k+1} &= \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{\Phi}_{k+1}\boldsymbol{M}_{k+1}^{-1}\boldsymbol{\Phi}_{k+1}^{\mathrm{T}})\boldsymbol{Y} \\ &\quad + 2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}_{k+1}\boldsymbol{M}_{k+1}^{-1}\boldsymbol{\Psi}_{k+1}^{\mathrm{T}}\boldsymbol{E} \\ &\quad + \lambda\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{I} - \lambda\boldsymbol{\Psi}_{k+1}\boldsymbol{M}_{k+1}^{-1}\boldsymbol{\Psi}_{k+1}^{\mathrm{T}})\boldsymbol{E} \end{aligned} \tag{33}$$

where $\boldsymbol{M}_{k+1} = \boldsymbol{\Phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_{k+1} + \lambda\boldsymbol{\Psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_{k+1}$.

Then, the net contribution of the selected term $(\boldsymbol{\varphi}_i)$ is given as

$$\delta J_{k+1} = J_{k+1} - J_k \tag{34}$$

In order to simplify the calculations, three recursive matrices (indexed with (1), (2) and (3)) at $(k+1)$th iteration, recorded as $\boldsymbol{R}_{k+1}^{(1)}$, $\boldsymbol{R}_{k+1}^{(2)}$ and $\boldsymbol{R}_{k+1}^{(3)}$, are defined as

$$\boldsymbol{R}_{k+1}^{(1)} \triangleq \boldsymbol{I} - \boldsymbol{\Phi}_{k+1}\boldsymbol{M}_{k+1}^{-1}\boldsymbol{\Phi}_{k+1}^{\mathrm{T}} \tag{35}$$

$$\boldsymbol{R}_{k+1}^{(2)} \triangleq \boldsymbol{I} - \lambda \boldsymbol{\Psi}_{k+1} \boldsymbol{M}_{k+1}^{-1} \boldsymbol{\Psi}_{k+1}^{\mathrm{T}} \tag{36}$$

$$\boldsymbol{R}_{k+1}^{(3)} \triangleq -\boldsymbol{\Phi}_{k+1} \boldsymbol{M}_{k+1}^{-1} \boldsymbol{\Psi}_{k+1}^{\mathrm{T}} \tag{37}$$

where the matrix sizes of $\boldsymbol{R}_{k+1}^{(1)}$, $\boldsymbol{R}_{k+1}^{(2)}$ and $\boldsymbol{R}_{k+1}^{(3)}$ are all N by N, and $\boldsymbol{R}_0^{(1)} = \boldsymbol{I}_{N\times N}, \boldsymbol{R}_0^{(2)} = \boldsymbol{I}_{N\times N}, \boldsymbol{R}_0^{(3)} = \boldsymbol{0}_{N\times N}$.

Having defined these three recursive matrices, the explicit formulation to efficiently calculate the net contribution can be derived first. According to the calculation of $\boldsymbol{M}_k^{-1}$ using Equations (30) and (31), the recursive matrices defined in Equations (35), (36) and (37) can be recursively updated as follows

$$\boldsymbol{R}_{k+1}^{(1)} = \boldsymbol{R}_k^{(1)} - \frac{(\boldsymbol{R}_k^{(1)}\boldsymbol{\varphi}_i + \lambda \boldsymbol{R}_k^{(3)}\boldsymbol{\kappa}_i)(\boldsymbol{\varphi}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(1)})^{\mathrm{T}} + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(3)})^{\mathrm{T}})}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \times k = 0, \ldots, m-1, \quad i = k+1, \ldots, m. \tag{38}$$

$$\boldsymbol{R}_{k+1}^{(2)} = \boldsymbol{R}_k^{(2)} - \frac{(\boldsymbol{R}_k^{(2)}\boldsymbol{\kappa}_i + (\boldsymbol{R}_k^{(3)})^{\mathrm{T}}\boldsymbol{\varphi}_i)(\boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(2)})^{\mathrm{T}} + \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)})}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \times k = 0, \ldots, m-1, \quad i = k+1, \ldots, m. \tag{39}$$

$$\boldsymbol{R}_{k+1}^{(3)} = \boldsymbol{R}_k^{(3)} - \frac{(\boldsymbol{R}_k^{(1)}\boldsymbol{\varphi}_i + \lambda \boldsymbol{R}_k^{(3)}\boldsymbol{\kappa}_i)(\boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(2)})^{\mathrm{T}} + \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)})}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \times k = 0, \ldots, m-1, \quad i = k+1, \ldots, m. \tag{40}$$

The proof of these properties in Equations (38)–(40) is given in Appendix 2.

In addition, based on the definitions in Equation (35)–(37), several properties can be inferred as given in Equations (41)–(44).

$$\boldsymbol{R}_k^{(1)} = (\boldsymbol{R}_k^{(1)})^{\mathrm{T}}; \quad \boldsymbol{R}_k^{(2)} = (\boldsymbol{R}_k^{(2)})^{\mathrm{T}}, \quad k = 0, 1, \ldots, m-1. \tag{41}$$

$$\boldsymbol{R}_k^{(1)} = (\boldsymbol{R}_k^{(1)})^2 + \lambda \boldsymbol{R}_k^{(3)} (\boldsymbol{R}_k^{(3)})^{\mathrm{T}}; \quad k = 0, 1, \ldots, m-1. \tag{42}$$

$$\boldsymbol{R}_k^{(2)} = (\boldsymbol{R}_k^{(2)})^2 + (\boldsymbol{R}_k^{(3)})^{\mathrm{T}} \boldsymbol{R}_k^{(3)}; \quad k = 0, 1, \ldots, m-1. \tag{43}$$

$$\boldsymbol{R}_k^{(3)} = \boldsymbol{R}_k^{(1)} \boldsymbol{R}_k^{(3)} + \boldsymbol{R}_k^{(3)} \boldsymbol{R}_k^{(2)}; \quad k = 0, 1, \ldots, m-1. \tag{44}$$

Then, the cost function $\boldsymbol{J}_k$ at the kth iteration can be reformulated as

$$\boldsymbol{J}_k = \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{Y} + \lambda \boldsymbol{E}^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{E} + 2\lambda \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{E} + \lambda \boldsymbol{E}^{\mathrm{T}} \boldsymbol{E} \quad k = 0, \ldots, m-1. \tag{45}$$

And, the net contribution of the selected term ($\boldsymbol{\varphi}_i$) in Equation (34) can be rewritten as

$$\begin{aligned} \delta \boldsymbol{J}_{k+1}(\boldsymbol{\varphi}_i) &= \boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(1)} - \boldsymbol{R}_k^{(1)}) \boldsymbol{Y} \\ &\quad + \lambda \boldsymbol{E}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(2)} - \boldsymbol{R}_k^{(2)}) \boldsymbol{E} \\ &\quad + 2\lambda \boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(3)} - \boldsymbol{R}_k^{(3)}) \boldsymbol{E} \\ &\quad + (\lambda \boldsymbol{E}^{\mathrm{T}} \boldsymbol{E} - \lambda \boldsymbol{E}^{\mathrm{T}} \boldsymbol{E}) \\ &= \boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(1)} - \boldsymbol{R}_k^{(1)}) \boldsymbol{Y} \\ &\quad + \lambda \boldsymbol{E}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(2)} - \boldsymbol{R}_k^{(2)}) \boldsymbol{E} \\ &\quad + 2\lambda \boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_{k+1}^{(3)} - \boldsymbol{R}_k^{(3)}) \boldsymbol{E} \\ &\quad \times k = 0, \ldots, m-1. \end{aligned} \tag{46}$$

According to the properties given in Equations (38)–(39), Equation (46) can be calculated as follows

$$\begin{aligned} \delta \boldsymbol{J}_{k+1}(\boldsymbol{\varphi}_i) = &- \frac{\boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_k^{(1)}\boldsymbol{\varphi}_i + \lambda \boldsymbol{R}_k^{(3)}\boldsymbol{\kappa}_i)(\boldsymbol{\varphi}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(1)})^{\mathrm{T}} + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(3)})^{\mathrm{T}}) \boldsymbol{Y}}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \\ &- \lambda \frac{\boldsymbol{E}^{\mathrm{T}} (\boldsymbol{R}_k^{(2)}\boldsymbol{\kappa}_i + (\boldsymbol{R}_k^{(3)})^{\mathrm{T}}\boldsymbol{\varphi}_i)(\boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(2)})^{\mathrm{T}} + \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)}) \boldsymbol{E}}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \\ &- 2\lambda \frac{\boldsymbol{Y}^{\mathrm{T}} (\boldsymbol{R}_k^{(1)}\boldsymbol{\varphi}_i + \lambda \boldsymbol{R}_k^{(3)}\boldsymbol{\kappa}_i)(\boldsymbol{\kappa}_i^{\mathrm{T}} (\boldsymbol{R}_k^{(2)})^{\mathrm{T}} + \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)}) \boldsymbol{E}}{(\boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(1)} \boldsymbol{\varphi}_i + \lambda \boldsymbol{\kappa}_i^{\mathrm{T}} \boldsymbol{R}_k^{(2)} \boldsymbol{\kappa}_i + 2\lambda \boldsymbol{\varphi}_i^{\mathrm{T}} \boldsymbol{R}_k^{(3)} \boldsymbol{\kappa}_i)} \\ &\times k = 0, \ldots, m-1. \end{aligned} \tag{47}$$

4.1.2. Fast computation of net contributions of model terms to the cost function

Note that there are $m{-}k$ candidate terms (i.e. $i = k+1, \ldots, m$) to be evaluated at the $(k+1)$th iteration, and in order to select a new term $\boldsymbol{\phi}_{k+1}$ in $\boldsymbol{\Phi}_{k+1}$ to be included into the model, one needs to calculate $\delta \boldsymbol{J}_{k+1}(\varphi_i)$ $(i = k+1, \ldots, m)$ using (47) and the candidate term which is selected as $\boldsymbol{\phi}_{k+1}$ for inclusion into the model must have the maximal net contribution to the cost function, i.e.

$$\Delta \boldsymbol{J}_{k+1}(\phi_{k+1}) = \max_{\mathbf{Q}_{m-k}} \Delta \boldsymbol{J}_{k+1}(\varphi_i) \tag{48}$$

However, calculating the net contribution of $m-k$ candidate terms using (47) will consume a large amount of computing resources including memory and time, to further simplify the calculations, a few intermediate matrices $\boldsymbol{A}^{(l)} \triangleq [a_{j,i}^{(l)}] \in \Re^{m\times m}$, $\boldsymbol{b}^{(l)} \triangleq [b_i^{(l)}] \in \Re^{1\times m}$ $(l = 1, 2, 3, 4)$ are further introduced to calculate the net contribution to the cost function of the remaining $m-k$ candidates. For $\boldsymbol{A}^{(l)}$, the elements $a_{j,i}^{(l)}$ are given by:

$$a_{j,i}^{(1)} \triangleq \begin{cases} 0, & j > i \\ \boldsymbol{\phi}_j^{T}(\boldsymbol{R}_{j-1}^{(1)})\boldsymbol{\phi}_i, & j \le i \le k \\ \boldsymbol{\phi}_j^{\mathrm{T}}(\boldsymbol{R}_{j-1}^{(1)})\boldsymbol{\varphi}_i & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\varphi}_j^{\mathrm{T}}(\boldsymbol{R}_k^{(1)})\boldsymbol{\varphi}_i & \begin{array}{c} j = k+1, \ldots, m, \\ i = j, \ldots, m \end{array} \end{cases} \tag{49}$$

$$a_{j,i}^{(2)} \triangleq \begin{cases} 0, & j > i \\ \boldsymbol{\psi}_j^{T}(\boldsymbol{R}_{j-1}^{(2)})\boldsymbol{\psi}_i, & j \le i \le k \\ \boldsymbol{\psi}_j^{T}(\boldsymbol{R}_{j-1}^{(2)})\boldsymbol{\kappa}_i, & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\kappa}_j^{\mathrm{T}}(\boldsymbol{R}_k^{(2)})\boldsymbol{\kappa}_i & \begin{array}{c} j = k+1, \ldots, m, \\ i = j, \ldots, m \end{array} \end{cases} \tag{50}$$

$$a_{j,i}^{(3)} \triangleq \begin{cases} 0, & j > i \\ \boldsymbol{\psi}_j^{T}(\boldsymbol{R}_{j-1}^{(3)})\boldsymbol{\phi}_i, & j \le i \le k \\ \boldsymbol{\psi}_j^{T}(\boldsymbol{R}_{j-1}^{(3)})\boldsymbol{\varphi}_i, & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\kappa}_j^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})^{\mathrm{T}}\boldsymbol{\varphi}_i & \begin{array}{l} j = k+1, \ldots, m, \\ i = j, \ldots, m \end{array} \end{cases} \tag{51}$$

$$a_{j,i}^{(4)} \triangleq \begin{cases} 0, & j > i \\ \boldsymbol{\phi}_j^{T}(\boldsymbol{R}_{j-1}^{(3)})\boldsymbol{\psi}_i, & j \le i \le k \\ \boldsymbol{\phi}_j^{T}(\boldsymbol{R}_{j-1}^{(3)})\boldsymbol{\kappa}_i, & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\varphi}_j^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})\boldsymbol{\kappa}_i & \begin{array}{l} j = k+1, \ldots, m, \\ i = j, \ldots, m \end{array} \end{cases} \tag{52}$$

where the first three terms in each equation are used for recursive update of the elements, while the last term in each equation is used to calculate the net contribution of a candidate term and only the diagonal elements are used.

For $\boldsymbol{b}^{(l)}$, the elements $b_i^{(l)}$ for the remaining candidates at $(k+1)$th iteration are given by:

$$b_i^{(1)} \triangleq \begin{cases} \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{R}_{i-1}^{(1)})\boldsymbol{\phi}_i & i \le k \\ \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{R}_k^{(1)})\boldsymbol{\varphi}_i & i = k+1, \ldots, m \end{cases} \tag{53}$$

$$b_i^{(3)} \triangleq \begin{cases} \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{R}_{i-1}^{(3)})\boldsymbol{\psi}_i & i \le k \\ \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})\boldsymbol{\kappa}_i & i = k+1, \ldots, m \end{cases} \tag{54}$$

$$b_i^{(2)} \triangleq \begin{cases} \boldsymbol{E}^{\mathrm{T}}(\boldsymbol{R}_{i-1}^{(2)})\boldsymbol{\psi}_i & i \le k \\ \boldsymbol{E}^{\mathrm{T}}(\boldsymbol{R}_k^{(2)})\boldsymbol{\kappa}_i & i = k+1, \ldots, m \end{cases} \tag{55}$$

$$b_i^{(4)} \triangleq \begin{cases} \boldsymbol{E}^{\mathrm{T}}(\boldsymbol{R}_{i-1}^{(3)})^{\mathrm{T}}\boldsymbol{\phi}_i & i \le k \\ \boldsymbol{E}^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})^{\mathrm{T}}\boldsymbol{\varphi}_i & i = k+1, \ldots, m \end{cases} \tag{56}$$

where the first term in each equation is used for recursive update of the elements, while the last term in each equation is used to calculate the net contribution.

Because of the symmetrical property and the properties in Equations (41)–(44), it is evident that $a_{j,i}^{(1)} = (a_{j,i}^{(1)})^{\mathrm{T}}$, $a_{j,i}^{(2)} = (a_{j,i}^{(2)})^{\mathrm{T}}$, $b_i^{(1)} = (b_i^{(1)})^{\mathrm{T}}$ and $b_i^{(3)} = (b_i^{(3)})^{\mathrm{T}}$.

Using Equation (38) and the definition of $a_{j,i}^{(1)}$ in Equation (49), $a_{j,i}^{(1)}$ can be recursively computed. Taking the second term of (49) as an example, it can be recursively updated as follows

$$\begin{aligned} a_{j,i}^{(1)} &= \boldsymbol{\phi}_j^{\mathrm{T}}(\boldsymbol{R}_{j-1}^{(1)})\boldsymbol{\phi}_i = \boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{R}_{j-2}^{(1)}\boldsymbol{\phi}_i \\ &\quad - \frac{\begin{array}{c} \boldsymbol{\phi}_j^{\mathrm{T}}(\boldsymbol{R}_{j-2}^{(1)}\boldsymbol{\phi}_{j-1} + \lambda \boldsymbol{R}_{j-2}^{(3)}\boldsymbol{\psi}_{j-1}) \\ (\boldsymbol{\phi}_{j-1}^{\mathrm{T}}(\boldsymbol{R}_{j-2}^{(1)})^{\mathrm{T}} + \lambda \boldsymbol{\psi}_{j-1}^{\mathrm{T}}(\boldsymbol{R}_{j-2}^{(3)})^{\mathrm{T}})\boldsymbol{\phi}_i \end{array}}{\begin{array}{c} (\boldsymbol{\phi}_{j-1}^{\mathrm{T}}\boldsymbol{R}_{j-2}^{(1)}\boldsymbol{\phi}_{j-1} + \lambda \boldsymbol{\psi}_{j-1}^{\mathrm{T}}\boldsymbol{R}_{j-2}^{(2)}\boldsymbol{\psi}_{j-1} \\ +2\lambda \boldsymbol{\phi}_{j-1}^{\mathrm{T}}\boldsymbol{R}_{j-2}^{(3)}\boldsymbol{\psi}_{j-1}) \end{array}} \\ &= \boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{R}_{j-2}^{(1)}\boldsymbol{\phi}_i - \frac{(a_{j-1,j}^{(1)} + \lambda a_{j-1,j}^{(3)})(a_{j-1,i}^{(1)} + \lambda a_{j-1,i}^{(3)})}{a_{j-1,j-1}^{(1)} + \lambda a_{j-1,j-1}^{(2)} + 2\lambda a_{j-1,j-1}^{(3)}} \end{aligned} \tag{57}$$

This updating procedure based on (38) will continue until the recursive matrix reaches the values $\boldsymbol{R}_0^{(1)}$, then

$$a_{j,i}^{(1)} = \boldsymbol{\phi}_j^{\mathrm{T}}\boldsymbol{\phi}_i - \sum_{s=1}^{j} \frac{(a_{s,j}^{(1)} + \lambda a_{s,j}^{(3)})(a_{s,i}^{(1)} + \lambda a_{s,i}^{(3)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}}$$

$$j \le i \le k \tag{58}$$

By adopting the similar recursive procedure to the other terms in (38), $a^{(1)}_{j,i}$ can be calculated as follows

$$a^{(1)}_{j,i} = \begin{cases} \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{\phi}_i - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le i \le k \\ \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{\varphi}_i - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\varphi}_j^{\mathrm{T}} \boldsymbol{\varphi}_i - \displaystyle\sum_{s=1}^{k} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j = k+1, \ldots, m, \quad i = j, \ldots, m \end{cases} \tag{59}$$

Similarly, $a^{(2)}_{j,i}$, $a^{(3)}_{j,i}$, $a^{(4)}_{j,i}$, $b^{(1)}_i$, $b^{(3)}_i$, $b^{(2)}_i$ and $b^{(4)}_i$ can be quickly calculated as follows using the properties Equations (38)–(40).

$$a^{(2)}_{j,i} = \begin{cases} \boldsymbol{\psi}_j^{\mathrm{T}} \boldsymbol{\psi}_i - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le i \le k \\ \boldsymbol{\psi}_j^{\mathrm{T}} \boldsymbol{\kappa}_i - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le k, \quad i = k+1, \ldots, m \\ \boldsymbol{\kappa}_j^{\mathrm{T}} \boldsymbol{\kappa}_i - \displaystyle\sum_{s=1}^{k} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j = k+1, \ldots, m, \quad i = j, \ldots, m \end{cases} \tag{60}$$

$$a^{(3)}_{j,i} = \begin{cases} - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le i \le k \\ - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le k, \quad i = k+1, \ldots, m \\ - \displaystyle\sum_{s=1}^{k} \frac{(a^{(2)}_{s,j} + a^{(4)}_{s,j})(a^{(1)}_{s,i} + \lambda a^{(3)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j = k+1, \ldots, m, \quad i = j, \ldots, m \end{cases} \tag{61}$$

$$a^{(4)}_{k+1,i} = \begin{cases} - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le i \le k \\ - \displaystyle\sum_{s=1}^{j-1} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j \le k, \quad i = k+1, \ldots, m \\ - \displaystyle\sum_{s=1}^{k} \frac{(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})(a^{(2)}_{s,i} + a^{(4)}_{s,i})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & j = k+1, \ldots, m, \quad i = j, \ldots, m \end{cases} \tag{62}$$

$$b^{(1)}_{i} = \begin{cases} \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\phi}_i - \displaystyle\sum_{s=1}^{j-1} \frac{(b^{(1)}_{s} + \lambda b^{(3)}_{s})(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & i \le k \\ \boldsymbol{Y}^{\mathrm{T}} \boldsymbol{\varphi}_i - \displaystyle\sum_{s=1}^{k} \frac{(b^{(1)}_{s} + \lambda b^{(3)}_{s})(a^{(1)}_{s,j} + \lambda a^{(3)}_{s,j})}{a^{(1)}_{s,s} + \lambda a^{(2)}_{s,s} + 2\lambda a^{(3)}_{s,s}} & i = k+1, \ldots, m \end{cases} \tag{63}$$

$$b_i^{(3)} = \begin{cases} -\sum_{s=1}^{j-1} \dfrac{(b_s^{(1)} + \lambda b_s^{(3)})(a_{s,k+1}^{(2)} + a_{s,k+1}^{(4)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i \le k \\ -\sum_{s=1}^{k} \dfrac{(b_s^{(1)} + \lambda b_s^{(3)})(a_{s,k+1}^{(2)} + a_{s,k+1}^{(4)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i = k+1, \dots, m \end{cases} \tag{64}$$

$$b_i^{(2)} = \begin{cases} \boldsymbol{E}^{\mathrm{T}} \boldsymbol{\psi}_i - \sum_{s=1}^{j-1} \dfrac{(b_s^{(2)} + b_s^{(4)})(a_{s,j}^{(2)} + a_{s,j}^{(4)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i \le k \\ \boldsymbol{E}^{\mathrm{T}} \boldsymbol{\kappa}_i - \sum_{s=1}^{k} \dfrac{(b_s^{(2)} + b_s^{(4)})(a_{s,j}^{(2)} + a_{s,j}^{(4)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i = k+1, \dots, m \end{cases} \tag{65}$$

$$b_i^{(4)} = \begin{cases} -\sum_{s=1}^{j-1} \dfrac{(b_s^{(2)} + b_s^{(4)})(a_{s,j}^{(1)} + \lambda a_{s,j}^{(3)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i \le k \\ -\sum_{s=1}^{k} \dfrac{(b_s^{(2)} + b_s^{(4)})(a_{s,j}^{(1)} + \lambda a_{s,j}^{(3)})}{a_{s,s}^{(1)} + \lambda a_{s,s}^{(2)} + 2\lambda a_{s,s}^{(3)}} & i = k+1, \dots, m \end{cases} \tag{66}$$

Noticing from Equations (47) and (49)–(56), the net contribution of any candidate term $\boldsymbol{\varphi}_i$ $(i = k+1, \dots, m)$ at $(k+1)$th iteration can be quickly calculated as follows

$$\begin{aligned} \delta \boldsymbol{J}_{k+1}(\boldsymbol{\varphi}_i) = {} & \frac{(b_i^{(1)} + \lambda b_i^{(3)})^2}{a_{i,i}^{(1)} + \lambda a_{i,i}^{(2)} + 2\lambda a_{i,i}^{(3)}} \\ & + \frac{\lambda (b_i^{(2)} + b_i^{(4)})^2}{a_{i,i}^{(1)} + \lambda a_{i,i}^{(2)} + 2\lambda a_{i,i}^{(3)}} \\ & + \frac{2\lambda (b_i^{(1)} + \lambda b_i^{(3)})(b_i^{(2)} + b_i^{(4)})}{a_{i,i}^{(1)} + \lambda a_{i,i}^{(2)} + 2\lambda a_{i,i}^{(3)}} \end{aligned} \tag{67}$$

Equation (67) gives a simplified formulation for calculating the net contribution with low demand on computation and memory resources. Accordingly, the regressor with the maximum net contribution will be selected from the candidates based on their net contributions $\delta \boldsymbol{J}_{k+1}(\boldsymbol{\varphi}_i)$. The selected term will then be moved to the $(k+1)$th position of the matrix $\boldsymbol{\Phi}_{k+1}$ and denoted as $\boldsymbol{\phi}_{k+1}$.

4.1.3. Fast computation of model coefficients

The next step is then to develop effective formula for estimating coefficients. According to the definition (28), the optimal model coefficients (26) can be rewritten as

$$\hat{\boldsymbol{\Theta}}_k = \boldsymbol{M}_k^{-1} (\boldsymbol{\Phi}_k^{\mathrm{T}} \boldsymbol{Y} - \lambda \boldsymbol{\Psi}_k^{\mathrm{T}} \boldsymbol{E}) \tag{68}$$

To use (35) and (37) to simplify the calculations, both sides of (68) are multiplied by $\boldsymbol{\Phi}_k$, yielding

$$\begin{aligned} \boldsymbol{\Phi}_k \hat{\boldsymbol{\Theta}}_k &= \boldsymbol{\Phi}_k \boldsymbol{M}_k^{-1} (\boldsymbol{\Phi}_k^{\mathrm{T}} \boldsymbol{Y} - \lambda \boldsymbol{\Psi}_k^{\mathrm{T}} \boldsymbol{E}) \\ &= (\boldsymbol{I} - \boldsymbol{R}_k^{(1)}) \boldsymbol{Y} + \lambda \boldsymbol{R}_k^{(3)} \boldsymbol{E} \end{aligned} \tag{69}$$

Expanding $\boldsymbol{\Phi}_k \hat{\boldsymbol{\Theta}}_k$ in Equation (69) gives:

$$\hat{\theta}_1 \boldsymbol{\phi}_1 + \cdots + \hat{\theta}_k \boldsymbol{\phi}_k = (\boldsymbol{I} - \boldsymbol{R}_k^{(1)}) \boldsymbol{Y} + \lambda \boldsymbol{R}_k^{(3)} \boldsymbol{E} \tag{70}$$

Referring to Li et al. (2005), a recursive matrix at $(k+1)$th iteration is defined as $\boldsymbol{R}_{k+1}^{(0)} \triangleq \boldsymbol{I} - \boldsymbol{\Phi}_{k+1} (\boldsymbol{\Phi}_{k+1}^{\mathrm{T}} \boldsymbol{\Phi}_{k+1})^{-1} \boldsymbol{\Phi}_{k+1}^{\mathrm{T}}$ (where $\boldsymbol{R}_0^{(0)} = \boldsymbol{I}_{N \times N}$) for parameter estimation using the backward substitution.

Then, both sides of Equation (70) are multiplied by $(\boldsymbol{R}_{j-1}^{(0)} \boldsymbol{\phi}_j)^{\mathrm{T}}$ and consider the property $\boldsymbol{R}_k^{(0)} \boldsymbol{\phi}_i = 0$ $(i < k)$ (Li et al., 2005), yielding

$$\begin{aligned} & \sum_{i=j}^{k} \hat{\theta}_i (\boldsymbol{R}_{j-1}^{(0)} \boldsymbol{\phi}_j)^{\mathrm{T}} \boldsymbol{\phi}_i \\ & \quad = \sum_{i=j}^{k} \hat{\theta}_i \boldsymbol{\phi}_j^{\mathrm{T}} \boldsymbol{R}_{j-1}^{(0)} \boldsymbol{\phi}_i \\ & \quad = (\boldsymbol{R}_{j-1}^{(0)} \boldsymbol{\phi}_j)^{\mathrm{T}} ((\boldsymbol{I} - \boldsymbol{R}_k^{(1)}) \boldsymbol{Y} + \lambda \boldsymbol{R}_k^{(3)} \boldsymbol{E}) \\ & \quad = (\boldsymbol{R}_{j-1}^{(0)} \boldsymbol{\phi}_j)^{\mathrm{T}} (\boldsymbol{Y} - \boldsymbol{R}_k^{(1)} \boldsymbol{Y} + \lambda \boldsymbol{R}_k^{(3)} \boldsymbol{E}) \quad j = 1, \dots, k. \end{aligned} \tag{71}$$

For the final model consisting of $M = h + r - 1$ regressors, Equation (70) can be rewritten as

$$\sum_{i=j}^{M}\hat{\theta}_i(\boldsymbol{R}^{(0)}_{j-1}\boldsymbol{\phi}_j)^{\mathrm{T}}\boldsymbol{\phi}_i = (\boldsymbol{R}^{(0)}_{j-1}\boldsymbol{\phi}_j)^{\mathrm{T}} \times (\boldsymbol{Y} - \boldsymbol{R}^{(1)}_M\boldsymbol{Y} + \lambda\boldsymbol{R}^{(3)}_M\boldsymbol{E}), \quad j = 1,\ldots,M. \tag{72}$$

Obviously, the model coefficients of the last selected term can be first calculated using (72) and then the others can be backward recursively computed. However, the computation is still high due to the presence of $\boldsymbol{R}^{(0)}_{j-1}$, $\boldsymbol{R}^{(1)}_k$ and $\boldsymbol{R}^{(3)}_k$. To simplify the calculations, a recursive matrix $\boldsymbol{A} \triangleq [a_{k+1,i}] \in \Re^{M\times M}$ and three recursive vectors $\boldsymbol{Y}^{(k)} \in \Re^{N\times 1}$, $\boldsymbol{E}^{(k)} \in \Re^{N\times 1}$ and $\boldsymbol{b} \triangleq [b_{k+1}] \in \Re^{1\times M}$ are defined as

$$\boldsymbol{Y}^{(k)} \triangleq \boldsymbol{R}^{(1)}_k\boldsymbol{Y} \tag{73}$$

$$\boldsymbol{E}^{(k)} \triangleq \boldsymbol{R}^{(3)}_k\boldsymbol{E} \tag{74}$$

$$a_{k+1,i} \triangleq \boldsymbol{\phi}^{\mathrm{T}}_{k+1}\boldsymbol{R}^{(0)}_k\boldsymbol{\phi}_i \tag{75}$$

$$b_{k+1} \triangleq (\boldsymbol{Y} - \boldsymbol{Y}^{(M)} + \lambda\boldsymbol{E}^{(M)})^{\mathrm{T}}\boldsymbol{R}^{(0)}_k\boldsymbol{\phi}_{k+1} \tag{76}$$

where $k = 0, 1, \ldots, M-1$ representing the number of selected model terms and $i = k+1, \ldots, M$ is the index of the term whose coefficient has been estimated.

For the selected model terms at $(k+1)$th time, Equation (72) can be rewritten as

$$\sum_{i=k+1}^{M}\hat{\theta}_i a_{k+1,i} = (\boldsymbol{R}^{(0)}_k\boldsymbol{\phi}_{k+1})^{\mathrm{T}}(\boldsymbol{Y} - \boldsymbol{Y}^{(\boldsymbol{M})} + \lambda\boldsymbol{E}^{(M)}) = b_{k+1} \tag{77}$$

To further simplify $\boldsymbol{Y}^{(M)}$ and $\boldsymbol{E}^{(M)}$, two recursive regressors $\boldsymbol{\phi}^{(k)}_i$, $\boldsymbol{\psi}^{(k)}_i$ are defined as follows:

$$\boldsymbol{\phi}^{(k)}_i \triangleq \boldsymbol{R}^{(1)}_k\boldsymbol{\phi}_i \tag{78}$$

$$\boldsymbol{\psi}^{(k)}_i \triangleq \boldsymbol{R}^{(3)}_k\boldsymbol{\psi}_i \tag{79}$$

where $k = 0, 1, \ldots, M-1$ and $i = 1, 2, \ldots, k$.

To select M model terms, the output $\boldsymbol{Y}$ and the regressors $\boldsymbol{\phi}_i$, $\boldsymbol{\psi}_i$, $\boldsymbol{E}$ will be recursively updated for M times. Their updated notations are defined as $\boldsymbol{Y}^{(M)}$, $\boldsymbol{\phi}^{(M)}_i$, $\boldsymbol{\psi}^{(M)}_i$ and $\boldsymbol{E}^{(M)}$ respectively. Based on Equations (38), (40) (73), (74), (78) and (79), the four vectors are recursively updated by $\boldsymbol{R}^{(1)}_M$ or $\boldsymbol{R}^{(3)}_M$. Similar to the derivation of $a^{(1)}_{k+1,i}$, the four vectors can be calculated as

$$\boldsymbol{\phi}^{(M)}_i = \boldsymbol{\phi}_i - \sum_{j=1}^{M}\frac{(\boldsymbol{\phi}^{(j-1)}_j + \lambda\boldsymbol{\psi}^{(j-1)}_j)(a^{(1)}_{j,i} + \lambda a^{(3)}_{j,i})}{a^{(1)}_{j,j} + \lambda a^{(2)}_{j,j} + 2\lambda a^{(3)}_{j,j}} \tag{80}$$

$$\boldsymbol{\psi}^{(M)}_i = -\sum_{j=1}^{M}\frac{(\boldsymbol{\phi}^{(j-1)}_j + \lambda\boldsymbol{\psi}^{(j-1)}_j)(a^{(2)}_{j,i} + a^{(4)}_{j,i})}{a^{(1)}_{j,j} + \lambda a^{(2)}_{j,j} + 2\lambda a^{(3)}_{j,j}} \tag{81}$$

Thus

$$\boldsymbol{\phi}^{(M)}_i + \lambda\boldsymbol{\psi}^{(M)}_i = \boldsymbol{\phi}_i - \sum_{j=1}^{M}\frac{(a^{(1)}_{j,i} + \lambda a^{(2)}_{j,i} + \lambda a^{(3)}_{j,i} + \lambda a^{(4)}_{j,i})(\boldsymbol{\phi}^{(j-1)}_j + \lambda\boldsymbol{\psi}^{(j-1)}_j)}{a^{(1)}_{j,j} + \lambda a^{(2)}_{j,j} + 2\lambda a^{(3)}_{j,j}} \tag{82}$$

And then

$$\boldsymbol{Y}^{(M)} = \boldsymbol{Y} - \sum_{j=1}^{M}\frac{(\boldsymbol{\phi}^{(j-1)}_j + \lambda\boldsymbol{\psi}^{(j-1)}_j)(b^{(1)}_j + \lambda b^{(3)}_j)}{a^{(1)}_{j,j} + \lambda a^{(2)}_{j,j} + 2\lambda a^{(3)}_{j,j}} \tag{83}$$

$$\boldsymbol{E}^{(M)} = -\sum_{j=1}^{M}\frac{(\boldsymbol{\phi}^{(j-1)}_j + \lambda\boldsymbol{\psi}^{(j-1)}_j)(b^{(2)}_j + b^{(4)}_j)}{a^{(1)}_{j,j} + \lambda a^{(2)}_{j,j} + 2\lambda a^{(3)}_{j,j}} \tag{84}$$

For the definitions of $\boldsymbol{R}^{(0)}_k$, Equations (75) and (76) are similar as those defined in Li et al. (2005), and $a_{k+1,i}$ and b_{k+1} can be calculated in a similar way as for the calculation of $a^{(1)}_{k+1,i}$ using

$$a_{k+1,i} = \boldsymbol{\phi}^{\mathrm{T}}_{k+1}\boldsymbol{\phi}_i - \sum_{j=1}^{k}\frac{(a_{j,k+1}a_{j,i})}{a_{j,j}} \tag{85}$$

$$b_{k+1} = (\boldsymbol{Y} - \boldsymbol{Y}^{(M)} + \lambda\boldsymbol{E}^{(M)})^{\mathrm{T}}\Phi_{k+1} - \sum_{j=1}^{k}\frac{(b_j a_{j,k+1})}{a_{j,j}} \tag{86}$$

Accordingly, after updating these above matrices and vectors in M iterations, Equation (77) can be reformulated as follows:

$$\hat{\theta}_j = \left(b_j - \sum_{i=j+1}^{M}\hat{\theta}_i a_{j,i}\right)/a_{j,j} \quad j = M, M-1, \ldots, 1. \tag{87}$$

Remark 1: To identify a fraction model using the proposed RFRA, Equation (67) is first used for model term selection, which computes the net contribution of a candidate regression term to the cost function, then Equation (87) is used for model coefficient estimation once the model terms have been selected.

4.2. Regularisation analysis and the proposed algorithm

In this paper, a regularisation term is used to compensate the estimation bias which is induced by the linear transformation. The value of the regularisation parameter has to be carefully chosen, as inappropriate value will either lead to the overfitting or underfitting problem. According to Proposition 3.1, the regularisation parameter needs to satisfy Equation (23). Furthermore, Equation (23) is based on known model structure. However, the structure is unknown a prior, therefore, the regularisation parameter has to be identified in two steps, including the selection of the magnitude order of the regularisation parameter and value refinement during the process.

The proposed method is detailed in Algorithm 1:

Algorithm 1 The regularised FRA algorithm for fraction model identification

```
Require: the input matrix $\boldsymbol{x} \in \Re^{N\times q}$, the output vector $\boldsymbol{Y} = \boldsymbol{y} \in \Re^{N\times 1}$, the candidate base function vector $\mathbf{Q}_m \in \Re^{m\times 1}$, the maximum total number of the denominator terms and the numerator terms $M$.
Ensure: $h$ selected terms form the numerator regression matrix $\boldsymbol{P}_n \in \Re^{h\times N}$, $r$ selected terms form the denominator regression matrix $\boldsymbol{P}_d \in \Re^{r\times N}$ and the first term $\boldsymbol{p}_{d1}$ of $\boldsymbol{P}_d$ is $\mathbf{1}_{1\times N}$, the coefficient of $\boldsymbol{\theta}_{d1}$ is 1, the selected regressors matrix of system (7), the coefficients $\hat{\boldsymbol{\Theta}} \in \Re^{M\times 1}$ and $r + h = M$.
Initialisation: form the candidate regression matrix $\mathbf{Q}_m = [\boldsymbol{\varphi}_1(\boldsymbol{x}), \cdots, \boldsymbol{\varphi}_m(\boldsymbol{x})]$, set the regulation parameters $\lambda$, the maximum number of the selected regressors $M$, $sse\_Y = \boldsymbol{Y}' * \boldsymbol{Y}$, $\boldsymbol{\Phi} = []$ to store the selected regressors for the model.
for $k = 1$ to $M - 1$ do
    1. calculate $a_{i,j}^{(1)}$, $a_{i,j}^{(2)}$, $a_{i,j}^{(3)}$, $a_{i,j}^{(4)}$, $b_j^{(1)}$, $b_j^{(2)}$, $b_j^{(3)}$ and $b_j^{(4)}$ $(i = 1, \cdots, k, j = 1, \cdots, m - k)$ using Equations (59)- -(63), Equations (65), (64) and (66).
    2. calculate the net contributions of all remaining candidate model terms $(\mathbf{Q}_{m-k})$ to the cost function based on Equation (67).
    3. select the model term producing the maximal net decrease to the cost function and save it in the matrix $\boldsymbol{\Phi}$ denoted as $\boldsymbol{\phi}_k$ and the remaining candidate model terms form a new candidate regression matrix $\mathbf{Q}_{m-k} = [\varphi_{k+1}, \cdots, \varphi_m]$.
    4. update $sse\_Y = sse\_Y - \delta \boldsymbol{J}_{k+1}$.
    if $sse\_Y < 0$ then
        break
    else
        continue
    end if
end for
coefficients Estimate
1. Calculate $\boldsymbol{\phi}_k^{(k-1)} + \lambda \boldsymbol{\psi}_k^{(k-1)}$, $\boldsymbol{Y}^{(M)}$ and $\boldsymbol{E}^{(M)}$ $(k = 1, \cdots, M)$ based on the selected matrix $\boldsymbol{\Phi}$ using Equations (82), (83) and (84).
2. Calculate $a_{i,k}$ $(i = k, \cdots, M)$ and $b_k$ $(k = 1, \cdots, M)$ using Equations (85) and (86).
3. Estimate the coefficients using Equation (87).
```

4.3. Computation complexity

The proposed algorithm, the orthogonal least sqare method (OLS) (Billings & Zhu, 1994) and the fast orthogonal identification method (Zhu & Billings, 1996) are all capable of simultaneously selecting the model terms and estimating the linear parameters. In Zhu and Billings (1996), it is shown that the fast orthogonal identification method is computationally more efficient than the conventional OLS method. In the following, the computational complexity of the proposed method will be compared with that of the fast orthogonal identification algorithm when both are applied to the identification problem including the structure detection and the parameters estimation.

Suppose there are initially m candidate nonlinear terms to build a fraction nonlinear dynamic model expressed in Equation (2), M terms $(M < m)$ will be selected to construct the model from the candidate terms and the coefficients of the resultant model should be estimated. The basic arithmetic operations used in the two identification algorithms are addition/subtraction, and multiplication/division. According to the procedures of RFRA in Algorithm 1, the model term selection process should repeat $M-1$ times and there are $m-k$ times for updating any of $a_{j,i}^{(1)}$, $a_{j,i}^{(2)}$, $a_{j,i}^{(3)}$, $a_{j,i}^{(4)}$, $b_i^{(1)}$, $b_i^{(2)}$, $b_i^{(3)}$, $b_i^{(4)}$ and the net contribution at each selection. Thus the operations in Equations (59)–(67) should be counted $m-k$ times at $(k+1)$th selection. In addition, Equations (82)–(87) only need to repeat M times because there are M coefficients for the selected terms. Thereby, for the proposed method, the total number of addition/subtraction and multiplication/division operations is:

$$\begin{aligned}C^{(RFRA)} &= 4NMm + 9mM(M-1) \\ &\quad - \frac{19M(M-1)(2M-1)}{6} \\ &\quad + 5M(2m-M) \\ &\quad - 2NM(M-1) + \frac{3M^2(M-1)}{2} \\ &\quad + 8NM - M - 4N \\ &\quad + \frac{3M(M-1)}{2} + 7mM - 5M^2 + NM^2 \\ &\quad - NM + 2M \end{aligned} \tag{88}$$

Referring to Zhu and Billings (1996), though there are also $M-1$ selections, additional calculations are however needed for the coefficients estimation and for computing the errors at each selection iteration. According to the implementation using Equation 5.4 in Zhu and Billings (1996), the total number of addition/subtraction and multiplication/division operations for the fast orthogonal method applied to the same model identification problem is:

$$\begin{aligned}C^{(OLS)} &= 6NMm - 2NM(M-1) + 8mM(M-1) \\ &\quad + 4Nm + 10NM - \frac{9M(M-1)}{2} - 4N \\ &\quad - \frac{13M(M-1)(2M-1)}{6} \\ &\quad + (M-1) + 22Mm \end{aligned} \tag{89}$$

The ratio of the computational complexity of the fast OLS algorithm in Zhu and Billings (1996) and the computational complexity of the proposed algorithm is illustrated in Figure 3 as a function of the number of candidate terms m and actual model size M. It is graphically shown that the proposed method is computationally more efficient than the fast OLS algorithm in all cases, and the efficiency is much higher as the model size decreases.

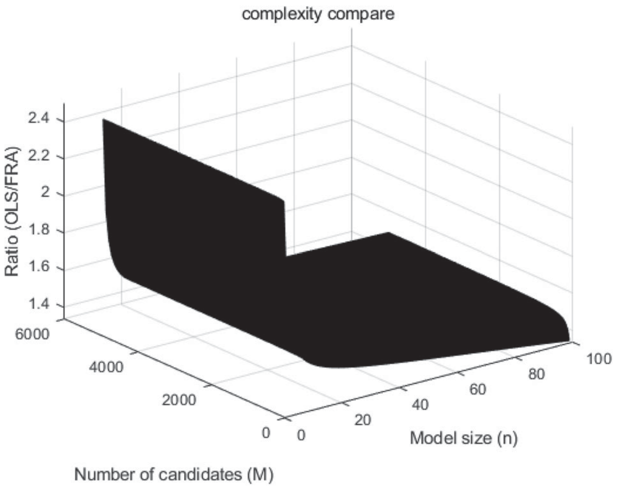

Figure 3. The ratio of computational complexities between the fast OLS method and the proposed method.

5. Simulation and analysis

To verify the effectiveness of the proposed method, two simulation examples are used. The first example is a rational model. The second one is the case where the regressors are not all polynomials. For each case, 1000 samples are generated with the same distribution of input and error signals.

The input signals for generating the training and testing samples are independent and uniformly distributed signals with amplitude range $[-1, 1]$ and variance of 0.33 and an independent zero mean Gaussian error with variance 0.01 is generated as used in Zhu and Billings (1996), S. Chen et al. (1989).

The proposed method is then applied to identify the system defined in Equation (90).

$$\begin{aligned}y(t) &= \frac{a(t)}{c(t)} + e(t) \\ &= \frac{\begin{array}{c}0.25y(t-1) - 0.85y(t-2)e(t-2) \\ +u(t-1)u(t-2)\end{array}}{1 + 0.68y^2(t-1) + y^2(t-2)} + e(t)\end{aligned} \tag{90}$$

The rational fraction system in Equation (90) is a typical rational nonlinear autoregressive moving average with eXogenous inputs (NARMAX) model (Waheeb et al., 2019). Models of this kind include past inputs, past outputs and past model errors which are the differences between the estimated values and the true values. Though incorporating the past error signal into the model may lead to a more accurate model, they are often difficult to measure or make

Table 1. Different regularisation parameters for Example 1.

Regularisation parameter	$-1e-5$	$-1e-9$	$1e{-9}$	$1e{-5}$
Selected term	$u(t-1)u(t-2)$	$u(t-1)u(t-2)$	$u(t-1)u(t-2)$	$u(t-1)u(t-2)$
	$y(t-1)$	$y(t-1)$	$y(t-1)$	$y(t-1)$
	$y(t)\cdot y^2(t-1)$	$y(t)\cdot y^2(t-2)$	$y(t)\cdot y^2(t-1)$	$y(t)\cdot y^2(t-2)$
	$y(t-2)u(t-2)$	$y(t)\cdot y^2(t-1)$	$y(t)\cdot y^2(t-2)$	$y(t)\cdot u^2(t-2)$
	$y(t-2)\varepsilon(t-2)$	$y(t-2)\varepsilon(t-2)$	$y(t)\cdot u^2(t-2)$	$y(t)\cdot u^2(t-1)$
Estimation parameters	0.9474	0.9977	0.9931	2.4071
	0.2359	0.2471	0.2460	0.4250
	0.6954	0.9722	0.6765	3.4022
	0.0322	0.6772	0.9941	0.0364
	−1.2269	−0.8541	−0.0125	2.6361
MSE of validation	6.5335e−04	9.6952e-05	1.0045e−04	0.0114

the modelling process rather complicated. It should be noted that errors always exist in modelling nonlinear systems (Nelles, 2020). The model error is often decomposed into the bias and the variance. The bias can be decreased to zero by increasing the model nonlinearity, while the variance is significantly influenced by the data size and data quality. Thus the error in Equation (90) is often treated as known a priori (Neji & Beji, 2000). In general, the errors are assumed as independent Gaussian random variables with zero mean and a finite variance, and the variance is often given a priori. To this end, in this case study, the variance is set to 0.01. Accordingly, 27 polynomial terms are generated first based on the principle that the highest order is 2 and the maximal delays of input, error and output signals are 2. According to Equation (7), the candidate terms consist of 54 polynomial regressors.

First, the regularisation parameter is symmetrically adjusted around 0 and a number of trials are tested. Table 1 shows four cases with different values for the regularisation parameter. It is found that −9 is the optimal magnitude order for the regularisation parameter in Example 1. Then, the regularisation parameter is fine tuned around the optimal magnitude order, and the final value for the regularisation parameter is chosen to be $1.8276e{-9}$.

Then the rational fraction system in Equation (90) is identified by the RFRA algorithm and the fast OLS method in reference Zhu and Billings (1996), respectively. The selected nonlinear terms from the above 54 nonlinear terms and their estimated linear parameters are listed in Table 2. By comparing the parameters obtained by the two algorithms, it is clear that the parameters identified by RFRA algorithm are closer to the actual parameters of the model, and the model obtained is more accurate than that produced by the fast OLS method in reference Zhu and Billings (1996).

Table 2. Comparison of estimation coefficients for Example 1.

Selection order using RFRA	Selected term	True Parameters	Estimated Para-meters using RFRA	Estimated Parameters using OLS
1	$u(t-1)\cdot u(t-2)$	1	0.9977	0.9973
2	$y(t-1)$	0.25	0.2471	0.269
3	$y(t)\cdot y^2(t-2)$	1	0.9722	0.963
4	$y(t)\cdot y^2(t-1)$	0.68	0.6772	0.654
5	$y(t-2)\cdot e(t-2)$	−0.85	−0.8541	−0.613

Finally, a more general fraction system given in Equation (91), which contains both polynomials and other nonlinear terms in the rational function, is used as the second case study.

$$y(t)=\frac{a(t)}{c(t)}+e(t)$$
$$=\frac{\begin{array}{c}0.3y(t-1)-0.7y(t-2)\\+\sin(u(t-1))+u(t-1)\end{array}}{1+\exp(-y^2(t-1))+u^2(t-2)}+e(t) \tag{91}$$

Similarly, nonlinear candidate items are generated. Considering the small number of samples which is only 1000, different basis functions are used as the numerator and denominator candidate terms, leading to the generation of 52 candidate terms.

It is shown that the bias becomes greater in the positive direction in Example 1, hence, similarly the regularisation parameter is fine tuned along the negative values for Example 2. Following the same procedure in example 1, the regularisation parameter is firstly coarsely adjusted and Table 3 shows some cases with different values for the regularisation parameter. After numerous trials, it is evident again that −9 is the optimum magnitude order for the regularisation parameter. Then, the regularisation parameter is further fine tuned. It is noted that the magnitude order of the MSE validation errors in all the two examples

Table 3. Different regularisation parameters for Example 2.

Regularisation parameter	$-1e-9$	$-1e-6$	0	1e–14
Selected term	$u(t-1)$ $y(t-2)$ $y(t-1)$ $\sin(u(t-1))$ $y(t)\cdot u^2(t-2)$ $y(t)\cdot$ $\exp(-y(t-1)^2)$	$u(t-1)$ $y(t-2)$ $y(t-1)$ $\sin(u(t-1))$ $y(t)\cdot u^2(t-2)$ $y(t)\cdot y(t-2)\cdot$ $u(t-2))$	$u(t-1)$ $y(t-2)$ $y(t-1)$ $\sin(u(t-1))$ $y(t)\cdot u^2(i-2)$ $y(t)\cdot \exp(-u(t-1)\cdot$ $u(t-2))$	$u(t-1)$ $y(t-2)$ $y(t-1)$ $\sin(u(t-1))$ $y(t)\cdot u^2(t-2)$ $y(t)\cdot \exp(-u(t-1)\cdot$ $u(t-2))$
Estimation Parameters	0.9359 −0.6642 0.2825 0.9068 0.8935 0.9597	0.5795 −0.3586 0.1573 0.1828 −0.0553 0.4437	1.5871 −0.5697 0.243 0.7952 0.7864 −0.2129	1.5871 −0.5697 0.243 0.7952 0.7863 −0.2129
MSE of Validation	1.0693e–04	0.0016	1.3552e–04	1.3552e–04

Table 4. Structure selection and parameters estimation for Example 2.

Selection order	Selected term	True parameters	Estimated parameters
1	$u(t-1)$	1	0.9611
2	$y(t-2)$	−0.7	−0.6728
3	$y(t-1)$	0.3	0.2871
4	$u^2(t-2)$	1	0.9299
5	$\exp(-y^2(t-1))$	1	0.9210
6	$\sin(u(t-1))$	1	0.9618

could achieve to −4 even without fine-tuning. The nonlinear regressors are selected and the coefficients are estimated which are shown in Table 4. It is again evident that the fraction system can be well identified using the proposed method.

6. Conclusion

This paper has proposed a novel fast method to identify the fraction models where the model structure and the coefficients are simultaneously estimated. The identification of the fraction system is first transformed to an equivalent quadratic optimisation problem by adding a regularisation term which is formulated as the inner product of the denominator terms. To solve the quadratic optimisation problem, a regularised fast recursive algorithm is developed. The proposed method is further validated by two simulation examples to confirm its efficacy. The method can be further applied to build the output layer of the traditional neural networks or the full connected layer of the deep neural networks which can reduce the size of the neural networks. However, a potential limitation of the proposed method is the computational effort to fine tune the regularisation parameter though we have shown that it value has to be negative and can not exceed $\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}$. It will inevitably cost more computations to optimise the regularisation parameter if the system is too complex. Fast optimisation of the penalty parameter will be a future research topic.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by Key Project of Science and Technology Commission of Shanghai Municipality [No. 19500712300], 111 Project [No. D18003].

Data availability statement

The authors confirm that the data supporting the findings of this study are available within the article.

Notes on contributors

Li Zhang received the B.Sc. and M.Sc. degrees from Anhui University of Technology, Anhui, China, in 2009 and 2013, respectively. She is currently pursuing the Ph.D. degree in control theory and control engineering at Shanghai University. Between 2018 and 2019, she was a visiting Ph.D. student with the University of Leeds, U.K. Her current research interests include nonlinear system modeling and identification, machine learning algorithm and power system.

Kang Li (M'05-SM'11) received the B.Sc. degree in Industrial Automation from Xiangtan University, Hunan, China, in 1989, the M.Sc. degree in Control Theory and Applications from Harbin Institute of Technology, Harbin, China, in 1992, and the Ph.D. degree in Control Theory and Applications from Shanghai Jiaotong University, Shanghai, China, in 1995. He also received D.Sc. degree in system identification from Queen's University Belfast, UK, in 2015. Between 1995 and 2002, he worked at Shanghai Jiaotong University, Delft University of Technology and Queen's University Belfast as a research fellow.

Between 2002 and 2018, he was a Lecturer (2002), a Senior Lecturer (2007), a Reader (2009) and a Chair Professor (2011) with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, U.K. He currently holds the Chair of Smart Energy Systems at the University of Leeds, UK. His research interests cover nonlinear system modelling, identification, and control, and machine learning, with substantial applications to energy and power systems, smart grid, electric vehicles, railway systems, and energy management in energy intensive manufacturing processes. He has authored/co-authored over 200 journal publications and edited/co-edited 19 conference proceedings.
Dr Li chaired the IEEE UKRI Control and Communication Ireland chapter, and was the Secretary of the IEEE UK & Ireland Section.

Dajun Du received the B.Sc. degree in electrical technology, and M.Sc. degrees in control theory and control engineering all from the Zhengzhou University, China in 2002 and 2005, respectively, and his Ph. D. degree in control theory and control engineering from Shanghai University in 2010. From September 2008 to September 2009, he was a visiting PhD student at Queen's University Belfast, UK. From April 2011 to August 2012, he was a Research Fellow at Queen's University Belfast, UK. He is currently a professor in Shanghai University. His main research interests include system modelling and identification and networked control systems.

Yihuan Li received the B.Sc. and M.Sc. degrees from North China Electric Power University, Beijing, China, in 2014 and 2017, respectively, and the PhD degree from University of Leeds in 2021. She is currently a lecture in North China Electric Power University. Her main research interests include system identification and state estimation, and the application of machine learning techniques in the field of battery management.

Minrui Fei received his B.S. and M.S. degrees in Industrial Automation from the Shanghai University of Technology in 1984 and 1992, respectively, and his PhD degree in Control Theory and Control Engineering from Shanghai University in 1997. Since 1998, he has been a full professor at Shanghai University. He is Chairman of Embedded Instrument and System Sub-society, and Standing Director of China Instrument & Control Society; Chairman of Life System Modeling and Simulation Sub-society, Vice-chairman of Intelligent Control and Intelligent Management Sub-society, and Director of Chinese Artificial Intelligence Association. His research interests are in the areas of networked control systems, intelligent control, complex system modeling, hybrid network systems, and field control systems.

ORCID

Dajun Du http://orcid.org/0000-0003-2979-1507

References

Andreev, M., & Borulko, V. (2017). Polynomial and rational approximate models of microwave elements. In 2017 XXIIND International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED) (pp. 256–259). IEEE.

Austin, A. P., Krishnamoorthy, M., Leyffer, S., Mrenna, S., Mller, J., & Schulz, H. (2021). Practical algorithms for multivariate rational approximation. *Computer Physics Communications*, *261*, 107663. https://doi.org/10.1016/j.cpc.2020.107663.

Bai, E.-w., Li, K., Zhao, W., & Xu, W. (2014). Kernel based approaches to local nonlinear non-parametric variable selection. *Automatica*, *50*(1), 100–113. https://doi.org/10.1016/j.automatica.2013.10.010

Billings, S., & Mao, K. (1998). Structure detection for nonlinear rational models using genetic algorithms. *International Journal of Systems Science*, *29*(3), 223–231. https://doi.org/10.1080/00207729808929516

Billings, S., & Zhu, Q. (1991). Rational model identification using an extended least-squares algorithm. *International Journal of Control*, *54*(3), 529–546. https://doi.org/10.1080/00207179108934174

Billings, S., & Zhu, Q. (1994). A structure detection algorithm for nonlinear dynamic rational models. *International Journal of Control*, *59*(6), 1439–1463. https://doi.org/10.1080/00207179408923140

Birpoutsoukis, G., Marconato, A., Lataire, J., & Schoukens, J. (2017). Regularized nonparametric volterra kernel estimation. *Automatica*, *82*, 324–327. https://doi.org/10.1016/j.automatica.2017.04.014

Castella, M., Pesquet, J.-C., & Marmin, A. (2018). Rational optimization for nonlinear reconstruction with approximate ℓ_0 penalization. *IEEE Transactions on Signal Processing*, *67*(6), 1407–1417. https://doi.org/10.1109/TSP.2018.2890065

Chen, J., Liu, Y., & Zhu, Q. (2018). Bias compensation recursive algorithm for dual-rate rational models. *IET Control Theory & Applications*, *12*(16), 2184–2193. https://doi.org/10.1049/cth2.v12.16

Chen, S., Billings, S. A., & Luo, W. (1989). Orthogonal least squares methods and their application to non-linear system identification. *International Journal of Control*, *50*(5), 1873–1896. https://doi.org/10.1080/00207178908953472

Chen, T., Andersen, M. S., Ljung, L., Chiuso, A., & Pillonetto, G. (2014). System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques. *IEEE Transactions on Automatic Control*, *59*(11), 2933–2945. https://doi.org/10.1109/TAC.2014.2351851

Curtain, R., & Morris, K. (2009). Transfer functions of distributed parameter systems: A tutorial. *Automatica*, *45*(5), 1101–1116. https://doi.org/10.1016/j.automatica.2009.01.008

Deng, J., Li, K., Harkin-Jones, E., Fei, M., & Li, S. (2013). A novel two stage algorithm for construction of RBF neural models based on a-optimality criterion. In 2013 Ninth International Conference On Natural Computation (ICNC) (pp. 1–7). IEEE.

Deng, J., Li, K., & Irwin, G. W. (2010). A two-stage algorithm for automatic construction of RBF neural models. In *Melecon 2010 15th IEEE Mediterranean Electrotechnical Conference* (pp. 166–171). IEEE.

Deng, J., Li, K., & Irwin, G. W. (2012). Locally regularised two-stage learning algorithm for rbf network centre selection. *International Journal of Systems Science*, *43*(6), 1157–1170. https://doi.org/10.1080/00207721.2010.545490

Du, D., Li, X., Fei, M., & Irwin, G. W. (2012). A novel locally regularized automatic construction method for rbf neural models. *Neurocomputing*, *98*, 4–11. https://doi.org/10.1016/j.neucom.2011.05.045.

Du, D., Zhu, M., Li, X., Fei, M., Bu, S., Wu, L., & Li, K. (2022). A review on cybersecurity analysis, attack detection, and attack defense methods in cyber-physical power systems. *Journal of Modern Power Systems and Clean Energy, to be published.* https://doi.org/10.35833/MPCE.2021.000604.

Gambella, C., Ghaddar, B., & Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, *290*(3), 807–828. https://doi.org/10.1016/j.ejor.2020.08.045

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, *36*(3), 1171–1220. https://doi.org/10.1214/009053607000000677

Hong, X., Mitchell, R., Chen, S., Harris, C., Li, K., & Irwin, G. (2008). Model selection approaches for non-linear system identification: A review. *International Journal of Systems Science*, *39*(10), 925–946. https://doi.org/10.1080/00207720802083018

Hussein, A. A.-h., & Batarseh, I. (2011). An overview of generic battery models. In *2011 IEEE Power And Energy Society General Meeting* (pp. 1–6). IEEE.

Jin, R., & Weng, G. (2019). A robust active contour model driven by pre-fitting bias correction and optimized fuzzy c-means algorithm for fast image segmentation. *Neurocomputing*, *359*, 408–419. https://doi.org/10.1016/j.neucom.2019.06.019

Kambhampati, C., Mason, J., & Warwick, K. (2000). A stable one-step-ahead predictive control of non-linear systems. *Automatica*, *36*(4), 485–495. https://doi.org/10.1016/S0005-1098(99)00173-9

Li, K. (2005). Eng-genes: A new genetic modelling approach for nonlinear dynamic systems. *IFAC Proceedings Volumes*, *38*(1), 162–167. https://doi.org/10.3182/20050703-6-CZ-1902.01105

Li, K., & Peng, J.-X. (2007). Neural input selectiona fast model-based approach. *Neurocomputing*, *70*(4), 762–769. https://doi.org/10.1016/j.neucom.2006.10.011

Li, K., Peng, J.-X., & Bai, E.-W. (2006). A two-stage algorithm for identification of nonlinear dynamic systems. *Automatica*, *42*(7), 1189–1197. https://doi.org/10.1016/j.automatica.2006.03.004

Li, K., Peng, J.-X., & Irwin, G. W. (2005). A fast nonlinear model identification method. *IEEE Transactions on Automatic Control*, *50*(8), 1211–1216. https://doi.org/10.1109/TAC.2005.852557

Liang, Z., & Zhang, L. (2021). Support vector machines with the epsilon-insensitive pinball loss function for uncertain data classification. *Neurocomputing*, *457*, 117–127. https://doi.org/10.1016/j.neucom.2021.06.044

Luo, X., Chang, X., & Ban, X. (2016). Regression and classification using extreme learning machine based on l1-norm and l2-norm. *Neurocomputing*, *174*, 179–186. https://doi.org/10.1016/j.neucom.2015.03.112

Mall, R., & Suykens, J. A. K. (May 2015). Very sparse LSSVM reductions for large-scale data. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(5), 1086–1097. https://doi.org/10.1109/TNNLS.2014.2333879

Manela, B., & Biess, A. (2021). Bias-reduced hindsight experience replay with virtual goal prioritization. *Neurocomputing*, *451*, 305–315. https://doi.org/10.1016/j.neucom.2021.02.090

Münker, T., & Nelles, O. (2016). Nonlinear system identification with regularized local fir model networks. *IFAC-PapersOnLine*, *49*(5), 61–66. https://doi.org/10.1016/j.ifacol.2016.07.090

Neji, Z., & Beji, F.-M. (2000). Neural network and time series identification and prediction. In *Proceedings Of The IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives For The New Millennium* (Vol. 4, p. 461–466). IEEE.

Nelles, O. (2020). *Nonlinear system identification, from classical approaches to neural networks, fuzzy models, and gaussian processes*. Springer.

Nielsen, M. (2019). Neural networks and deep learning. Retrieved from http://neuralnetworksanddeeplearning.com/chap2.html (Online Book).

Ohlsson, H., Ljung, L., & Boyd, S. (2010). Segmentation of arx-models using sum-of-norms regularization. *Automatica*, *46*(6), 1107–1111. https://doi.org/10.1016/j.automatica.2010.03.013

Paduart, J., Lauwers, L., Swevers, J., Smolders, K., Schoukens, J., & Pintelon, R. (2010). Identification of nonlinear systems using polynomial nonlinear state space models. *Automatica*, *46*(4), 647–656. https://doi.org/10.1016/j.automatica.2010.01.001

Piga, D., & Tóth, R. (2014). A bias-corrected estimator for nonlinear systems with outputerror type model structures. *Automatica*, *50*(9), 2373–2380. https://doi.org/10.1016/j.automatica.2014.07.021

Pillonetto, G. (2018). System identification using kernel-based regularization: New insights on stability and consistency issues. *Automatica*, *93*, 321–332. https://doi.org/10.1016/j.automatica.2018.03.065

Pillonetto, G., & De Nicolao, G. (2010). A new kernel-based approach for linear system identification. *Automatica*, *46*(1), 81–93. https://doi.org/10.1016/j.automatica.2009.10.031

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., & Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, *50*(3), 657–682. https://doi.org/10.1016/j.automatica.2014.01.001

Qiao, J., Meng, X., & Li, W. (2018). An incremental neuronal-activity-based rbf neural network for nonlinear system modeling. *Neurocomputing*, *302*, 1–11. https://doi.org/10.1016/j.neucom.2018.01.001

Senhaji, K., Ramchoun, H., & Ettaouil, M. (2020). Training feedforward neural network via multiobjective optimization model using non-smooth l1/2 regularization. *Neurocomputing*, *410*, 1–11. https://doi.org/10.1016/j.neucom.2020.05.066

Singh, J., & Dhiman, G. (2021). A survey on machine-learning approaches: Theory and their concepts. *Materials Today: Proceedings*. Advance online publication. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214785321039298.

Suykens, J. A., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing*, *48*(1), 85–105. https://doi.org/10.1016/S0925-2312(01)00644-0

Swarnakar, J., Sarkar, P., Dey, M., & Singh, L. J. (2017, Dec). Rational approximation of fractional order system in delta domain a unified approach. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 144–150). IEEE.

Tao, J., Wen, S., & Hu, W. (2015). L1-norm locally linear representation regularization multi-source adaptation learning. *Neural Networks*, *69*, 80–98. https://doi.org/10.1016/j.neunet.2015.01.009

Tikhonov, A. N., & Glasko, V. B. (1965). Use of the regularization method in non-linear problems. *USSR Computational Mathematics and Mathematical Physics*, *5*(3), 93–107. https://doi.org/10.1016/0041-5553(65)90150-3

Torres-Barrán, A., Alaíz, C. M., & Dorronsoro, J. R. (2018). ν-svm solutions of constrained lasso and elastic net. *Neurocomputing*, *275*, 1921–1931. https://doi.org/10.1016/j.neucom.2017.10.029

Valencia, E. A., & lvarez, M. A. (2017). Short-term time series prediction using hilbert space embeddings of autoregressive processes. *Neurocomputing*, *266*, 595–605. https://doi.org/10.1016/j.neucom.2017.05.067

Waheeb, W., Ghazali, R., & Shah, H. (2019). Nonlinear autoregressive moving-average (NARMA) time series forecasting using neural networks. In 2019 International Conference on Computer and Information Sciences (ICCIS) (pp. 1–5).IEEE.

Wang, L., Liu, J., & Qian, F. (2021). Wind speed frequency distribution modeling and wind energy resource assessment based on polynomial regression model. *International Journal of Electrical Power and Energy Systems*, *130*, 106964. https://doi.org/10.1016/j.ijepes.2021.106964

Yang, C., Zhu, X., Qiao, J., & Nie, K. (2020). Forward and backward input variable selection for polynomial echo state networks. *Neurocomputing*, *398*, 83–94. https://doi.org/10.1016/j.neucom.2020.02.034

Yang, Z., Mourshed, M., Liu, K., Xu, X., & Feng, S. (2020). A novel competitive swarm optimized rbf neural network model for short-term solar power generation forecasting. *Neurocomputing*, *397*, 415–421. https://doi.org/10.1016/j.neucom.2019.09.110

Yildiz, B., Bilbao, J., & Sproul, A. (2017). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, *73*, 1104–1122. https://doi.org/10.1016/j.rser.2017.02.023

Zhang, J., Li, K., Irwin, G. W., & Zhao, W. (2012). A regression approach to LS-SVM and sparse realization based on fast subset selection. In 2012 10th World Congress on Intelligent Control and Automation (WCICA) (pp. 612–617). IEEE.

Zhang, L., Li, K., Bai, E.-W., & Irwin, G. W. (2015). Two-stage orthogonal least squares methods for neural network construction. *IEEE Transactions on Neural Networks and Learning Systems*, *26*(8), 1608–1621. https://doi.org/10.1109/TNNLS.2014.2346399

Zhang, L., Zhang, D., Sun, J., Wei, G., & Bo, H. (2019). Salient object detection by local and global manifold regularized svm model. *Neurocomputing*, *340*, 42–54. https://doi.org/10.1016/j.neucom.2019.02.041

Zhao, W., Zhang, J., & Li, K. (2015). An efficient ls-svm-based method for fuzzy system construction. *IEEE Transactions on Fuzzy Systems*, *23*(3), 627–643. https://doi.org/10.1109/TFUZZ.2014.2321594

Zhu, Q. (2003). A back propagation algorithm to estimate the parameters of non-linear dynamic rational models. *Applied Mathematical Modelling*, *27*(3), 169–187. https://doi.org/10.1016/S0307-904X(02)00097-5

Zhu, Q. (2005). An implicit least squares algorithm for nonlinear rational model parameter estimation. *Applied Mathematical Modelling*, *29*(7), 673–689. https://doi.org/10.1016/j.apm.2004.10.008

Zhu, Q., & Billings, S. (1996). Fast orthogonal identification of nonlinear stochastic models and radial basis function neural networks. *International Journal of Control*, *64*(5), 871–886. https://doi.org/10.1080/00207179608921662

Zhu, Q., Wang, Y., Zhao, D., Li, S., & Billings, S. A. (2015). Review of rational (total) nonlinear dynamic system modelling, identification, and control. *International Journal of Systems Science*, *46*(12), 2122–2133. https://doi.org/10.1080/00207721.2013.849774

Zhu, Q., Yu, D., & Zhao, D. (2017). An enhanced linear Kalman filter (enlkf) algorithm for parameter estimation of nonlinear rational models. *International Journal of Systems Science*, *48*(3), 451–461. https://doi.org/10.1080/00207721.2016.1186243

Zhu, Q., Zhang, W., Zhang, J., & Sun, B. (2019). U-neural network-enhanced control of nonlinear dynamic systems. *Neurocomputing*, *352*, 12–21. https://doi.org/10.1016/j.neucom.2019.04.008

Appendices

Appendix 1. Simplifying the regularized cost function

According to the principles of matrix manipulation, the cost function is simplified in Equation (A1).

$$
\begin{aligned}
J &= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta} + \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Theta}\\
&\quad + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta} + \lambda\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}\boldsymbol{\Theta}\\
&= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})\\
&\quad + (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\\
&\quad \times \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})\\
&\quad + \lambda(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\\
&\quad \times \boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})\\
&\quad + \lambda(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}) - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}\\
&= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad + (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})\\
&\quad \times (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})\\
&\quad + \lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} - 2\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + 4\lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi} - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi})(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y} - \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E})\\
&\quad - 2\lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} - 2\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + 4\lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad - 2\lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&= \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y} + \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} - \boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad + 2\lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{Y}\\
&\quad - \lambda\boldsymbol{E}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{E}\\
&= \boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}})\boldsymbol{Y}\\
&\quad + 2\lambda\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{E}\\
&\quad + \lambda\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{I} - \lambda\boldsymbol{\Psi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \lambda\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^{\mathrm{T}})\boldsymbol{E}
\end{aligned}
\tag{A1}
$$

Appendix 2. Proof of Properties of the recursive matrices

From the definition of M_k^{-1} in Equation (29), substituting Equations (30) and (31) to the definition of $R_{k+1}^{(1)}$ in Equation (35), the recursive formulation of $R_{k+1}^{(1)}$ is given in Equation (A2).

$$
\begin{aligned}
\boldsymbol{R}_k^{(1)} - \boldsymbol{R}_{k+1}^{(1)} &= \boldsymbol{\Phi}_{k+1}\boldsymbol{M}_{k+1}^{-1}\boldsymbol{\Phi}_{k+1}^{\mathrm{T}} - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&= \begin{bmatrix}\boldsymbol{\Phi}_k & \boldsymbol{\phi}_{k+1}\end{bmatrix}\begin{bmatrix}\boldsymbol{F}_k & \boldsymbol{g}_k\\ \boldsymbol{g}_k^{\mathrm{T}} & \boldsymbol{f}_k\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_k^{\mathrm{T}}\\ \boldsymbol{\phi}_{k+1}^{\mathrm{T}}\end{bmatrix} - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&= \boldsymbol{\Phi}_k\boldsymbol{F}_k\boldsymbol{\Phi}_k^{\mathrm{T}} + \boldsymbol{\phi}_{k+1}\boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{\Phi}_k^{\mathrm{T}} + \boldsymbol{\Phi}_k\boldsymbol{g}_k\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\\
&\quad + \boldsymbol{\phi}_{k+1}\boldsymbol{f}_k\boldsymbol{\phi}_{k+1}^{\mathrm{T}} - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&= \boldsymbol{\Phi}_k\boldsymbol{F}_k\boldsymbol{\Phi}_k^{\mathrm{T}} - \boldsymbol{\phi}_{k+1}\frac{\boldsymbol{\phi}_{k+1}\boldsymbol{\Phi}_k^{\mathrm{T}} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k}{\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}\boldsymbol{F}_k\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&\quad - \boldsymbol{\Phi}_k\boldsymbol{F}_k\frac{\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}{\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\\
&\quad + \boldsymbol{\phi}_{k+1}\frac{\begin{gathered}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\ +(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\\ \boldsymbol{F}_k(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\end{gathered}}{(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})^2}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\\
&\quad - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&= \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}} + \boldsymbol{\Phi}_k\frac{\begin{gathered}\boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\ (\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\end{gathered}}{\begin{gathered}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\ -(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\ (\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\end{gathered}}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&\quad - \frac{\begin{gathered}\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\\ \boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\ (\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\end{gathered}}{\begin{gathered}(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\ (\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\ -(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\ (\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\end{gathered}}\\
&\quad - \boldsymbol{\phi}_{k+1}\frac{\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k}{\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\
&\quad - \boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\frac{\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}{\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1} + \lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}
\end{aligned}
$$

$$
-\frac{\begin{array}{c}\boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\\(\boldsymbol{\phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\end{array}}
$$

$$
+\frac{\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\boldsymbol{\phi}_{k+1}^{\mathrm{T}}}{(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})^2}
$$

$$
+\boldsymbol{\phi}_{k+1}\frac{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\end{array}}{(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})^2}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}
$$

$$
+\frac{\begin{array}{c}\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1})^2\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\end{array}}
$$

$$
-\boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}=\boldsymbol{\Phi}_k\frac{\begin{array}{c}\boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\end{array}}{\begin{array}{c}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\end{array}}\boldsymbol{\Phi}_k^{\mathrm{T}}
$$

$$
-\frac{\begin{array}{c}\boldsymbol{\phi}_{k+1}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}\\-\boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\\boldsymbol{\phi}_{k+1}^{\mathrm{T}}+\boldsymbol{\phi}_{k+1}\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\end{array}}
$$

$$
=\frac{\begin{array}{c}(\boldsymbol{\Phi}_k\boldsymbol{M}_k^{-1}(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1})\\-\boldsymbol{\phi}_{k+1})((\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\\\boldsymbol{M}_k^{-1}\boldsymbol{\Phi}_k^{\mathrm{T}}-\boldsymbol{\phi}_{k+1}^{\mathrm{T}})\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\end{array}}
$$

$$
=\frac{\begin{array}{c}(\boldsymbol{R}_k^{(1)}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{R}_k^{(3)}\boldsymbol{\psi}_{k+1})\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}(\boldsymbol{R}_k^{(1)})^{\mathrm{T}}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})^{\mathrm{T}})\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\psi}_{k+1}\\-(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{\Phi}_k+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{\Psi}_k)\boldsymbol{M}_k^{-1}\\(\boldsymbol{\Phi}_k^{\mathrm{T}}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\Psi}_k^{\mathrm{T}}\boldsymbol{\psi}_{k+1}))\end{array}}
$$

$$
=\frac{\begin{array}{c}(\boldsymbol{R}_k^{(1)}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{R}_k^{(3)}\boldsymbol{\psi}_{k+1})\\(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}(\boldsymbol{R}_k^{(1)})^{\mathrm{T}}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}(\boldsymbol{R}_k^{(3)})^{\mathrm{T}})\end{array}}{\begin{array}{c}(\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{R}_k^{(1)}\boldsymbol{\phi}_{k+1}+\lambda\boldsymbol{\psi}_{k+1}^{\mathrm{T}}\boldsymbol{R}_k^{(2)}\boldsymbol{\psi}_{k+1}\\+2\lambda\boldsymbol{\phi}_{k+1}^{\mathrm{T}}\boldsymbol{R}_k^{(3)}\boldsymbol{\psi}_{k+1})\end{array}}\qquad\text{(A2)}
$$

Similarly, it can be proved that Equations (39) and (40) are indeed true.