Industrial Process Modelling with Mechanical Frequency Spectrum Data

Industrial Process Modelling with Mechanical Frequency Spectrum Data

By

Jian Tang and Wen Yu

Cambridge Scholars Publishing

Industrial Process Modelling with Mechanical Frequency Spectrum Data

By Jian Tang and Wen Yu

This book first published 2020

Cambridge Scholars Publishing

Lady Stephenson Library, Newcastle upon Tyne, NE6 2PA, UK

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN (10): 1-5275-4866-X
ISBN (13): 978-1-5275-4866-4

CONTENTS

PREFACE

The modeling of key parameters of production quality and quantity that are difficult to measure can be divided into first main methods and methods based on data. The first principal method constructs the energy or mass equation of the actual physical or chemical behavior of the industrial process to estimate process parameters that are difficult to measure. This is difficult to realize and cannot be successful when a complex and uncertain mechanism is used and this is especially true for mechanical vibration and acoustic signals. The data-based method can build a soft sensor model using knowledge from the limit domain expert more easily. This method has been widely used in different industrial processes. In this book, the authors have used a method that relies on mechanical frequency spectrum data.

Conventional time/frequency analytical tools, such as the fast Fourier transform and the wavelet transform, have been widely used for mechanical vibration and acoustic signals in order to obtain frequency spectrum data. Different methods of dimension reduction and soft data-controlled sensors, such as principal component analysis, neural networks, diffuse inference, and the support machine's vector, have been used to model the spectrum

data. Since rotating mechanical devices for industrial processes have different characteristics for mechanical vibration and acoustic signals, some new data-driven modeling techniques must be addressed. In this book, the authors have taken the soft sensor model based on mechanical rotation frequency spectrum data as a kind of small sample high-data modeling problem. This model will be constructed based on the characteristic analysis of the rotating mechanical device's acoustic and vibration signals. The time/frequency adapted decomposition technique will be used to obtain a multiple scale frequency spectrum. In fact, it is a process that involves the selective fusion of interesting information from multiple sources. In addition, the built model must be updated based on some new useful samples. Therefore, the soft sensor method, which is based on a single model, creates a traditional single model that uses subsets of characteristics and the learning parameters from the combinational optimized input frequency spectrum; the soft sensor method based on a selective set (SEN) selects different subsets of useful characteristics and samples of working conditions by simulating an expert cognitive process; and the soft online sensor method uses an approximate linear dependence condition to select the new samples to update both the feature extraction and the learning model. In this book, all modeling methods are validated according to the

mechanical vibration and acoustic signals of a practical device using a laboratory scale.

From 2009 to 2018, the authors began to study the modeling of industrial processes using data from the mechanical frequency spectrum. After 9 years of work, the result of the analysis of the mechanical signal's characteristics and their extraction was based on the vibration spectrum of a single scale. However, the extraction of characteristics from a spectrum of multiple scale frequencies was based on the nucleus, the adaptive extraction of the nucleus' latent characteristics, the characteristic of multiple sources of selective fusion using frequency spectrum at scale, selection of frequency spectral characteristics at multiple scales using sphere criteria, extraction of nonlinear frequency spectral characteristics using spectrum single-scale frequencies, sample multiple selective fusion conditions using a double-layer genetic algorithm based on a partial core set algorithm, and a double-layer SEN, which are all based on a mixed set construction strategy and core component analysis online models for process modeling. These results have been published in a variety of journals and conferences. The authors have gathered all these results within this book.

This book is organized in the form of a textbook for courses on the modeling processes of uncertain nonlinear systems. It could also be used as a self-learning tool. The expected level of competence for the reader is covered in non-linear systems analysis courses, statistical models, selective group learning, and uncertain numbers.

Many people have contributed to the form and substance of this book. The authors wish to express their sincere gratitude to their advisor, Professor Tianyou Chai, for his continued support, patience, motivation, enthusiasm, and immense knowledge. Professor Chai's guide helped them throughout their research and writing. In addition, we would like to thank Professor Xiaojie Zhou, Professor Liangyong Wang, Professor Meiying Jia, and Professor Jinliang Ding. Finally, the authors appreciate the time and dedication of their wives. Without them, this book would not have been possible.

Jian Tang and Wen Yu

Beijing China and Mexico City Mexico

May 2018

CHAPTER 1

INTRODUCTION

Some key production quality and quantity parameters are difficult to measure in heavy mechanical devices. However, it is essential to accurately measure these parameters in order to realize the operational optimization control of complex industrial processes [1]. For example, the accurate measurement of load parameters inside a ball mill can help realize the operational optimization of the mineral grinding process. However, the first principal models for measuring these process parameters are difficult to construct due to the complex working mechanism of these industrial processes. Moreover, the rotating working characteristics of these mechanical devices, such as ball mills, means that it is difficult to use direct measuring methods [2, 3]. Mechanical devices that use complex industrial processes produce strong vibrations and acoustic signals. However, in the time domain, valuable information relative to difficult-to-measure process parameters are buried in a wide-band random noise signal known as white noise [4]. Studies show that high-dimensional frequency spectra contain useful information for measuring these process parameters [5, 6, 7]. Thus, mechanical vibration and acoustic frequency spectra have been used for process monitoring and modeling [8, 9, 10], such as the load parameter modeling of a ball mill in the grinding process. In practical industrial processes, domain experts can efficiently monitor the process parameters of familiar mechanical devices by considering interesting information that has originated from different operating conditions and multiple sources. Thus,

the development of reliable online sensors has become one of the bottlenecks and challenges when simulating human cognitive behavior. A data-driven soft-sensing technique, which is one of the major solving methods, has been used in broad fields due to its inferential estimation capability [11,12]. This study only focuses on mechanical vibration and acoustic signals in order to model difficult-to-measure process parameters.

Theoretically, mechanical vibration and acoustic signals from industrial mechanical devices have non-stationary and multi-component characteristics. For example, a ball mill is a type of heavy high-energy-consuming mechanical equipment [13]. There are millions of balls inside the mill and they are arranged in different layers. These balls impact mineral ores and mill lines with different impact forces and periods, thereby producing multiple mechanical sub-signals with different time scales. The normal strong shell vibration signal is mixed with these multi-scale sub-signals. Shell vibration is one of the main sources of the measured acoustic signals near the mill-grinding zone. Domain experts can select useful multiple operating conditions and multi-source features to make final decisions. In the practical grinding process, experts always use acoustic signals. In nature, human ears act as sets of bandpass filters, which can discern useful information from multi-component signals. Moreover, the human brain has a multi-level structure. Simulating the expert cognitive process remains a difficult issue. Studies show that adopting signal processing techniques to analyze mechanical signals is one of the most commonly used strategies [14]. The transformed frequency spectra of mechanical vibration and acoustic signals contain useful information relative to difficult to measure process parameters [15]. Thus, in order to simulate expert cognitive process, at least three sub-problems, in terms of

selective fusion multi-condition samples and multi-source features, should be focused on time/frequency domain transformation, high-dimensional spectral feature reduction, and the soft measuring model.

1.1 Time/frequency decomposition

The time/frequency transform method uses the fast Fourier transform (FFT) algorithm for original mechanical vibration and acoustic signals. In [16], frequency spectra are considered to be single scale. However, these mechanical vibration and acoustic signals have non-stationary and multi-component characteristics. The FFT may be unsuitable for processing these mechanical signals [17]. Some time/frequency analysis methods, such as discrete wavelet transform, continuous wavelet transform, and wavelet packet transform, have been used for mechanical signal-based fault diagnosis [18, 10, 19, 20, 21, 22]. However, a suitable basic function has to be selected for any practical problem. Empirical mode decomposition (EMD) and its variants can overcome this problem by obtaining a set of intrinsic mode functions (IMFs) [23, 24] in terms of the frequency distribution of different sub-signals, which have been successfully used in different industrial processes [25, 26, 27, 28, 29]. These IMFs can be considered to provide multi-source information that represents different sensors [30].

1.1.1 Empirical mode decomposition (EMD)

Nonlinear and non-stationary data can be decomposed into IMFs using the EMD technique. The whole data must satisfy some assumptions [31]. The decomposed IMFs have two characteristics: (1) the numbers of extreme and zero-crossings must either be equal or different at most by one; (2) at any

point, the mean of the envelope defined by local maxima and local minima is zero. Therefore, EMD is an iterative sifting procedure. The procedure for the EMD method is an iterative sifting process.

The empirical mode decomposition (EMD) method can decompose a multi-component signal into IMFs with high to low frequencies [10]. An IMF is a sub-signal with a detailed physical meaning that satisfies the following two conditions: (1) in the whole dataset, the number of extrema and the number of zero-crossings must either be equal or differ at most by one; and (2), at any point, the mean value of the envelopes defined by local maxima and the envelope defined by the local minima must be zero. Normally, mechanical vibration and acoustical signals of the industrial devices have multi-component and non-stationary characteristics. The original EMD proposed by Huang et al. [10] decomposes the mechanical signal with the following steps:

Step (1): Find out the entire extremes of the original signal with length N, i.e., $\boldsymbol{x}^{\mathrm{t}} = \{x_n^{\mathrm{t}}\}_{n=1}^{N}$.

Step (2): Connect the local maximum and minimum values to the upper and lower envelopes.

Step (3): Calculate the mean value m_1 of the upper and lower envelopes, and then denote the difference between the $\boldsymbol{x}^{\mathrm{t}}$ and m_1 as the first component $\boldsymbol{h}_1$:

$$\boldsymbol{h}_1 = \boldsymbol{x}^{\mathrm{t}} - m_1 \tag{1.1}$$

Step (4): Check whether $\boldsymbol{h}_1$ satisfies IMF criteria.

Step (5): Ideally, if $\boldsymbol{h}_1$ is an IMF, then $\boldsymbol{h}_1$ is the first component of $\boldsymbol{x}^{\mathrm{t}}$. If not, repeat the above procedures from Step (1) to Step (3). Here, $\boldsymbol{h}_1$ is treated as the original signal,

$$\boldsymbol{h}_{11} = \boldsymbol{h}_1 - m_{11} \tag{1.2}$$

where m_{11} is the mean value of upper and lower envelopes of $\boldsymbol{h}_1$. This process can be repeated until $\boldsymbol{h}_{1k}$ becomes an IMF, that is

$$\boldsymbol{h}_{1k} = \boldsymbol{h}_{1(k-1)} - m_{1k} . \tag{1.3}$$

This component is taken as the first IMF, and then it is represented as

$$\boldsymbol{x}^{\mathrm{t}}_{\mathrm{EMD1}} = \boldsymbol{h}_{1k} . \tag{1.4}$$

Theoretically, $\boldsymbol{x}^{\mathrm{t}}_{\mathrm{EMD1}}$ has the shortest period: i.e., the sub-signal with the smallest time-scale.

Step (6): Separate $\boldsymbol{x}^{\mathrm{t}}_{\mathrm{EMD1}}$ from the original $\boldsymbol{x}^{\mathrm{t}}$,

$$\boldsymbol{r}^{\mathrm{t}}_{\mathrm{EMD1}} = \boldsymbol{x}^{\mathrm{t}} - \boldsymbol{x}^{\mathrm{t}}_{\mathrm{EMD1}} . \tag{1.5}$$

Here, $\boldsymbol{r}^{\mathrm{t}}_{\mathrm{EMDl}}$ is treated as the original signal and, by repeating the above processes, the second IMF component $\boldsymbol{x}^{\mathrm{t}}_{\mathrm{EMD2}}$ of $\boldsymbol{x}^{\mathrm{t}}$ can be obtained.

Step (7): Repeat the above process J times, then all the IMFs from $\boldsymbol{x}^{\mathrm{t}}$ will be obtained. As soon as $\boldsymbol{r}^{\mathrm{t}}_{\mathrm{EMD}J}$ becomes a monotonic function, the decomposition process can be stopped.

Finally, the EMD result can be represented as follows:

$$\boldsymbol{x}^{\mathrm{t}} = \sum_{j=1}^{J} \boldsymbol{x}_{\mathrm{EMD}j}^{\mathrm{t}} + \boldsymbol{r}_{\mathrm{EMD}J}^{\mathrm{t}} \ . \tag{1.6}$$

The flow chart of EMD technology [16] is shown in Figure 1-1.

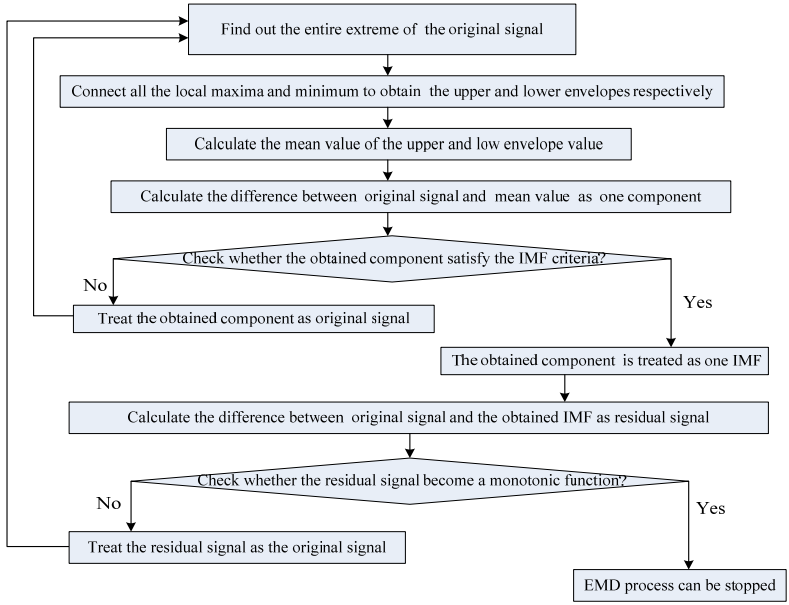

Figure 1-1 Flow chart of EMD technology

These IMFs are ordered from high frequency to low frequency with different time scales. The residual can either be a mean trend or a constant.

1.1.2 Ensemble EMD (EEMD)

There are several shortcomings in the original EMD method, such as lacking a theoretical foundation, end effects, sifting stop criterion, and extremum

interpolation [10]. The outstanding shortcoming of EMD is its mode mixing. An IMF with mode mixing can cease to possess physical meaning. Ensemble EMD (EEMD) can overcome this problem using noise-assisted analysis technology [10]. However, two decomposition parameters—e.g., amplitude of the added noise A_{noise} and the ensemble number M —should be selected. The influence of the two parameters can be described as follows:

$$\ln e_{\text{EEMD}} + \frac{A_{\text{noise}}}{2} \ln M = 0 \tag{1.7}$$

where e_{EEMD} represents the standard deviation of error between the original signal and the corresponding IMFs.

The decomposition steps of EEMD can be represented as (1) initialize M and A_{noise} ; (2) add A_{noise} into the original mechanical signal; (3) decompose the new signal with EMD using Step (1)–Step (7) and perform this M times; and, finally, (4) calculate the ensemble mean of EMD decomposition results of M times as the final EEMD result.

Finally, the EEMD result can be represented as follows:

$$\boldsymbol{x}^{\text{t}} = \sum_{j=1}^{J} \boldsymbol{x}_{\text{EEMD}j}^{\text{t}} + \boldsymbol{r}_{\text{EEMD}J}^{\text{t}} \tag{1.8}$$

The relation between EEMD and EMD can be represented as follows:

$$\begin{cases} \boldsymbol{x}_{\text{EEMD}j}^{\text{t}} = \dfrac{1}{M} \displaystyle\sum_{m=1}^{M} (\boldsymbol{x}_{\text{EMD}j}^{\text{t}})_m \\ \boldsymbol{r}_{\text{EEMD}J}^{\text{t}} = \dfrac{1}{M} \displaystyle\sum_{m=1}^{M} (\boldsymbol{r}_{\text{EMD}J}^{\text{t}})_m \end{cases} \tag{1.9}$$

where $(\boldsymbol{x}_{\text{EMD}j}^{\text{t}})_m$ represents the jth IMF of the mth EMD

decomposition and $(\boldsymbol{r}_{\mathrm{EMD}J}^{\mathrm{t}})_m$ represents its residual signal.

1.2 Small sample data modeling

Hardware sensors cannot measure many important process parameters in industrial processes, such as product quality. Nowadays, these variables mainly depend on manual timed sampling and titration in a chemical laboratory. Manual titration has a high precision, but the sampling interval is too long, and some parameters have to be judged by the experts' experience. They are not effective for real-time process monitoring and control. For example, the practical grinding process is continuous and ball mills are closely rotating working devices. Thus, useful shell vibration and acoustic modeling data can only be obtained during two phases: (1) the experimental design phase and (2) the stopping and re-starting phase of the ball mill. A sufficient number of well-represented training samples may be obtained at the expense of either economic benefits or a long waiting period. A large dataset cannot be established; therefore, research based on rotating mechanical frequency spectra must be conducted to address the problem of modeling high-dimensional, small samples. SVM employs the criterion that minimizes the structure risk, which is more prominent in the small-sample learning [32]. SVM needs to solve the quadratic program (QP) problem. Least square-support vector machines (LS-SVM) simplify the QP problem to solve a set of linear equations by changing the loss cost function in SVM to a sum-squared error [12].

The data in industrial processes are strongly coupled. It has been shown that the principal component analysis (PCA) can successfully monitor the changes of industrial processes [33], including chemical [34] and

microelectronics manufacturing processes [35]. It has been combined with many computational learning methods, such as artificial neural networks [36], neuro-fuzzy systems [37], support vector machines [38], fuzzy C-means [39], and multi-model technology [40]. However, extracted low dimensional independent features with PCA may have little relation to the predicted process parameters [41]. Projecting the latent structure or the partial least square (PLS) captures the maximal covariance between the two data blocks. This has been widely applied in chemometrics, steady-state process modeling, dynamic modeling, and process monitoring [42]. Some nonlinear approaches, such as quadratic PLS [43], neural network PLS [44], fuzzy PLS [45], and kernel PLS [46], are also useful in overcoming the disadvantages of PLS.

1.2.1 Support vector machines (SVM)

SVM is a training algorithm for learning classifications and regression rules from data. It uses regression theory to find a nonlinear map from the input space to the output space in order to map the data onto a higher dimensional feature space [40]. Therefore, SVM is a linear classifier in the parameter space, but it becomes a nonlinear classifier as a result of the nonlinear mapping of the space of the input patterns into the high dimensional feature space. Here, we have used SVM to build the soft sensor model between the feature spectral variable and the operating parameters of the ball mill.

We used the input/output data $[x(k), y(k)], k \in [1, l]$ to approximate a nonlinear function. Consider the following regression in a set of nonlinear functions:

$$f(x) = W^T \varphi(x) + b, \tag{1.10}$$

where the Kernel trick is $K(x, x_k) = \varphi(x)^T \varphi(x_k)$.

The cost function (empirical risk) is defined as

$$R_{emp}(\theta) = \frac{1}{l}\sum_{k=1}^{l} | y_k - (W^T \varphi(x) + b) |_{\varepsilon}. \tag{1.11}$$

where ε is the Vapnik's insensitive loss function. This is defined as

$$| y_k - f(x) |_{\varepsilon} = \begin{cases} 0 & | y_k - f(x) | \le \varepsilon, \\ | y_k - f(x) | - \varepsilon & otherwise, \end{cases} \tag{1.12}$$

Additionally, ε can be regarded as the accuracy of approximation.

The optimization problem is

$$\begin{aligned} &Min \qquad J_p = \frac{1}{2} W^T W, \\ &s.t: \qquad | y_k - (W^T \varphi(x_k) + b) | \le \varepsilon. \end{aligned} \tag{1.13}$$

If ε is too small, then certain points will be outside of the ε tube. Therefore, additional slack variables ξ_k and ξ_k^* are introduced. The optimization problem is modified into

$$\begin{aligned} &Min \qquad J_p = \frac{1}{2} W^T W + c\sum_{k=1}^{l} (\xi_k + \xi_k^*), \\ &s.t: \qquad -(\varepsilon + \xi_k^*) \le y_k - (W^T \varphi(x_k) + b) \le \varepsilon + \xi_k. \end{aligned} \tag{1.14}$$

The Lagrangian form is

$$\begin{aligned} L = J_p - \sum_{k=1}^{l} \alpha_k [(W^T x_k + b) - y_k + (\varepsilon + \xi_k)] - \\ \sum_{k=1}^{l} \alpha_k^* [(y_k - (W^T x_k + b) + (\varepsilon + \xi_k^*)] - \sum_{k=1}^{l} (\eta_k \xi_k + \eta_k^* \xi_k^*), \end{aligned} \tag{1.15}$$

where $\alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0$. The solution is the saddle point of the Lagrangian functional:

$$\max_{\alpha_k, \alpha_k^*, \eta_k, \eta_k^*} \quad \min_{W, b, \xi_k, \xi_k^* \min} L(W, b, \xi, \alpha, v). \tag{1.16}$$

The optimization problem in (13) can be changed into

$$\max_{\alpha, \alpha^*} \quad J_D(\alpha, \alpha^*) = -\frac{1}{2}\sum_{k=1}^{l}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)x_k^T X_l - \varepsilon\sum_{k=1}^{l}(\alpha_k - \alpha_k^*) + \sum_{k=1}^{l} y_k(\alpha_k - \alpha_k^*),$$
$$s.t: \quad \sum_{k=1}^{l}(\alpha_k - \alpha_k^*) = 0, \quad \alpha_k \geq 0. \tag{1.17}$$

The solution for α_k^* can be obtained by using a standard QP software package. The b can be solved by the following the Kuhn–Tucker conditions:

$$\alpha_k^*\{y_k[W^T\varphi(x) + b] - 1 + \xi_k\} = 0. \tag{1.18}$$

The resulting function is

$$f(x) = \sum_{k=1}^{l}(\alpha_k - \alpha_k^*)k(x_k, x) + b. \tag{1.19}$$

Since $\alpha_k^* = 0$, the solution vector is sparse. Therefore, the sum should only be taken over the non-zero α_k (support vector); this means the final result is

$$f(x) = \sum_{k=1}^{sv}(\alpha_k - \alpha_k^*)k(x_k, x) + b, \tag{1.20}$$

where sv is the number of support vectors.

1.2.2 Least square-support vector machines (LS-SVMs)

Given the training samples $\{\mathbf{X},\mathbf{y}\}=\{(\mathbf{x},y)_l\}_{l=1}^k$, we have the following function estimation problem

$$y(\mathbf{x})=\mathbf{W}^{\mathrm{T}}\Phi(\mathbf{x})+b \tag{1.21}$$

where $\Phi(*)$ maps $\{(\mathbf{x}_l\}_{l=1}^k$ onto a higher dimensional feature space; $\boldsymbol{W}$ is the weight vector; and b is the bias. The following problem will be solved:

$$\begin{cases}\min\limits_{W\cdot b} & J_{\mathrm{LSSVM}}=\dfrac{1}{2}\mathbf{W}^{\mathrm{T}}\mathbf{W}+\dfrac{1}{2}C_{\mathrm{LSSVM}}\sum\limits_{l=1}^{k}\xi_l^{\,2}\\ s.t: & y_l=\mathbf{W}^{T}\Phi(\mathbf{x})+b+\xi_l\end{cases} \tag{1.22}$$

where J_{LSSVM} is the objective function, C_{LSSVM} is the regularization parameter used to decide trade-off between model complexity and approximation accuracy, and ξ_l is the approximation error. The Lagrange from of Eq. (1.22) is

$$\begin{aligned}\mathrm{L}(\boldsymbol{W},b,\xi,\alpha)=&\frac{1}{2}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}+\frac{1}{2}C_{\mathrm{LSSVM}}\sum_{l=1}^{k}\xi_l^{\,2}\\&-\sum_{l=1}^{k}\alpha_l[\boldsymbol{W}^{\mathrm{T}}\Phi(\boldsymbol{z})+b+\xi_l-y_l]\end{aligned} \tag{1.23}$$

where α_l are Lagrange multipliers.

The solution is given by solving the following $(k+1)\times(k+1)$ linear equation:

$$\boldsymbol{A\Theta}=\boldsymbol{Y}' \tag{1.24}$$

where $\boldsymbol{A}=\begin{bmatrix}0 & \widetilde{1}^{\mathrm{T}}\\ \widetilde{1} & \boldsymbol{\Omega}+\dfrac{1}{C_{\mathrm{LSSVM}}}\boldsymbol{I}\end{bmatrix}$; $\boldsymbol{\Theta}=\begin{bmatrix}b\\ \boldsymbol{\alpha}\end{bmatrix}$, $\boldsymbol{Y}'=\begin{bmatrix}0\\ \boldsymbol{y}\end{bmatrix}$; $\widetilde{\boldsymbol{I}}=[1,1,\cdots,1]^{\mathrm{T}}$;

$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_k]^{\mathrm{T}}$; $\boldsymbol{y} = [y_1, y_2, \cdots, y_k]^{\mathrm{T}}$; $\boldsymbol{I}$ is a $k \times k$ identity matrix; and $\boldsymbol{\Omega}$ follows Mercer's condition.

The final result of the LSSVR models for ML parameters estimation is represented as

$$y(x_s) = \sum_{l=1}^{k} \beta_k \boldsymbol{K}(\boldsymbol{x}_s, \boldsymbol{x}_{s_l}) + b \,. \tag{1.25}$$

1.2.3 Project to latent structure or partial least square (PLS) [32]

Assume predictor variables $\mathbf{X} \in \Re^{n \times p}$ and response variables $\mathbf{Y} \in \Re^{n \times q}$ are normalized as $\mathbf{E}_0 = (\mathbf{E}_{01}\mathbf{E}_{02}...\mathbf{E}_{0p})_{n \times p}$ and $\mathbf{F}_0 = (\mathbf{F}_{01}\mathbf{F}_{02}...\mathbf{F}_{0q})_{n \times q}$, respectively. Let $\mathbf{t}_1$ be the first latent score vector of $\mathbf{E}_0$, $\mathbf{t}_1 = \mathbf{E}_0\mathbf{w}_1$, and $\mathbf{w}_1$ be the first axis of the $\mathbf{E}_0$, $\| \mathbf{w}_1 \| = 1$. Similarly, let $\mathbf{u}_1$ be the first latent score vector of $\mathbf{F}_0$, $\mathbf{u}_1 = \mathbf{F}_0\mathbf{c}_1$, and $\mathbf{c}_1$ be the first axis of the $\mathbf{F}_0$, $\| \mathbf{c}_1 \| = 1$. To maximize the covariance between $\mathbf{t}_1 = \mathbf{E}_0\mathbf{w}_1$ and $\mathbf{u}_1 = \mathbf{F}_0\mathbf{c}_1$, an optimization problem can be defined as

$$\mathrm{Max} < \mathbf{E}_0\mathbf{w}_1, \mathbf{F}_0\mathbf{c}_1 > \quad \text{s.t.} \quad \mathbf{w}_1{}^{\mathrm{T}}\mathbf{w}_1 = 1, \mathbf{c}_1{}^{\mathrm{T}}\mathbf{c}_1 = 1 \tag{1.26}$$

By solving (26) with the Lagrange approach, it is found that $\mathbf{w}_1$ and $\mathbf{c}_1$ are the maximum eigenvectors of matrix $\mathbf{E}_0^T\mathbf{F}_0\mathbf{F}_0{}^T\mathbf{E}_0$ and $\mathbf{F}_0{}^{\mathrm{T}}\mathbf{E}_0\mathbf{E}_0^{\mathrm{T}}\mathbf{F}_0$. Then, $\mathbf{t}_1$ and $\mathbf{u}_1$ are obtained. With $\mathbf{E}_0 = \mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{E}_1$, $\mathbf{F}_0 = \mathbf{u}_1\mathbf{q}_1^{\mathrm{T}} + \mathbf{F}_1^{0}$,

$\mathbf{F}_0 = \mathbf{t}_1\mathbf{b}_1^{\mathrm{T}} + \mathbf{F}_1$, $\mathbf{p}_1 = \dfrac{\mathbf{E}_0^{\mathrm{T}}\mathbf{t}_1}{\|\mathbf{t}_1\|^2}$, $\mathbf{q}_1 = \dfrac{\mathbf{F}_0^{\mathrm{T}}\mathbf{u}_1}{\|\mathbf{u}_1\|^2}$, and $\mathbf{b}_1 = \dfrac{\mathbf{F}_0^{\mathrm{T}}\mathbf{t}_1}{\|\mathbf{t}_1\|^2}$, the residual matrixes $\mathbf{E}_1$, $\mathbf{F}_1^0$, and $\mathbf{F}_1$ are obtained. By replacing $\mathbf{E}_0$ and $\mathbf{F}_0$ with $\mathbf{E}_1$ and $\mathbf{F}_1$, the second latent score vectors $\mathbf{t}_2$ and $\mathbf{u}_2$ can be obtained. Using the same procedure, all latent scores can be obtained until $\mathbf{E}_h = \mathbf{F}_h = 0$.

Therefore, PLS decomposes the data matrices $\mathbf{X}$ and $\mathbf{Y}$ into a low dimensional space with h latent variables (LVs), which can be shown as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{1.27}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{F} \tag{1.28}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_{\mathrm{h}}]$ and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_h]$ are the score matrices; $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_h]$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_h]$ are the loading matrices; and $\mathbf{E}$ and $\mathbf{F}$ are the modeling residuals of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

However, $\mathbf{T}$ cannot be calculated from the original $\mathbf{X}$ directly. First, denote $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{\mathrm{h}}]$, and let $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{\mathrm{h}}]$, where $\mathbf{r}_1 = \mathbf{w}_1$, for $i_{\mathrm{pls}} > 1$

$$\mathbf{r}_{i_{\mathrm{pls}}} = \prod_{j_{\mathrm{pls}}=1}^{i_{\mathrm{pls}}-1} \left(\mathbf{I}_p - \mathbf{w}_{j_{\mathrm{pls}}}\mathbf{p}_{j_{\mathrm{pls}}}^{\mathrm{T}}\right)\mathbf{w}_{i_{\mathrm{pls}}} \tag{1.29}$$

Thus, we have $\mathbf{R} = (\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$. Therefore, the score matrix $\mathbf{T}$ can be computed from original $\mathbf{X}$ as [47]

$$\mathbf{T} = \mathbf{X}\mathbf{R} = \mathbf{X}(\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1} \tag{1.30}$$

The regression model obtained from PLS algorithm can be represented as

$$\mathbf{Y} = \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{U}(\mathbf{T}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y} \tag{1.31}$$

The realization of the overall PLS algorithm is summarized as follows:

Algorithm PLS: Given matrix $\mathbf{X}$ and $\mathbf{Y}$,

Step 1: Scale the matrix $\mathbf{X}$ and $\mathbf{Y}$ to have a zero-unit variance.

Step 2: Let $E_0 = \mathbf{X}$, $F_0 = \mathbf{Y}$, and $h = 1$.

Step 3: For each LV h, take $u_h = y_{j_q}$, y_{j_q}, and some $\mathbf{F}_{h-1}$.

Step 4: Calculate the weights in matrix X: $\mathbf{w}_h^T = \mathrm{u}_h^T\mathbf{E}_{h-1}/(u_h^T u_h)$. Normalize $\mathbf{w}_{\mathrm{h}}$ to norm 1: $\mathbf{w}_h = \mathbf{w}_h / \| \mathbf{w}_h \|$.

Step 5: Calculate the X scores: $\mathbf{t}_h = \mathbf{E}_{h-1}\mathbf{w}_h$.

Step 6: Calculate the Y loadings: $\mathbf{q}_h^T = \mathbf{t}_h^T\mathbf{F}_{h-1}/(\mathbf{t}_h^T\mathbf{t}_h)$; normalize $\mathbf{q}_h$ to norm 1: $\mathbf{q}_h = \mathbf{q}_h / \|\mathbf{q}_h\|$.

Step 7: Calculate the Y scores: $\mathbf{u}_h = \mathbf{F}_{h-1}q_h$.

Step 8: Iterate Step 3–Step 6 until it converges. Compare the $\mathbf{t}_h$ in Step 5 with the one from the preceding iteration. If they are equal (within a certain rounding error), go to Step 9, and, if not, go to Step 4.

Step 9: Calculate the X loadings: $\mathbf{p}_h^T = \mathbf{t}_h^T\mathbf{E}_{h-1}/(\mathbf{t}_h^T\mathbf{t}_h)$. Normalize $\mathbf{p}_h = \mathbf{p}_h / \|\mathbf{p}_h\|$; $\mathbf{t}_h = \mathbf{t}_h \| \mathbf{p}_h \|$; and $\mathbf{w}_h = \mathbf{w}_h \| \mathbf{p}_h \|$.

Step 10: Calculate the regression factor: $\mathbf{b}_h^T = \mathbf{u}_h^T\mathbf{t}_{h-1}/(\mathbf{t}_h^T\mathbf{t}_h)$

Step 11: Calculate the residuals for LV h: $\mathbf{E}_h = \mathbf{E}_{h-1} - \mathbf{t}_h\mathbf{p}_h^T$; $\mathbf{F}_h = \mathbf{F}_{h-1} - b_h\mathbf{t}_h\mathbf{q}_h^T$.

Step 12: Let $h = h + 1$, return to Step 3 until all LVs are calculated.

1.2.4 Kernel PLS (KPLS)

The PLS algorithm constructs a linear multivariable regression model by extracting LVs from the original input/output data space. For high-dimensional linearity spectral data, the number of LVs is much lower than that of the original input features. Therefore, PLS is used widely in spectral data modeling. However, the application of PLS is limited by its linear assumption. In order to avoid PLS's shortcomings, there are two ways to construct it in a nonlinear manner: build a nonlinear inside model, or extend the nonlinear item to input it into the matrix $\mathbf{X}$. The KPLS algorithm belongs to the latter, which constructs a nonlinear model using the latent features in the nonlinear kernel feature space. The structure of KPLS algorithm [48] is shown in Figure 1-2.

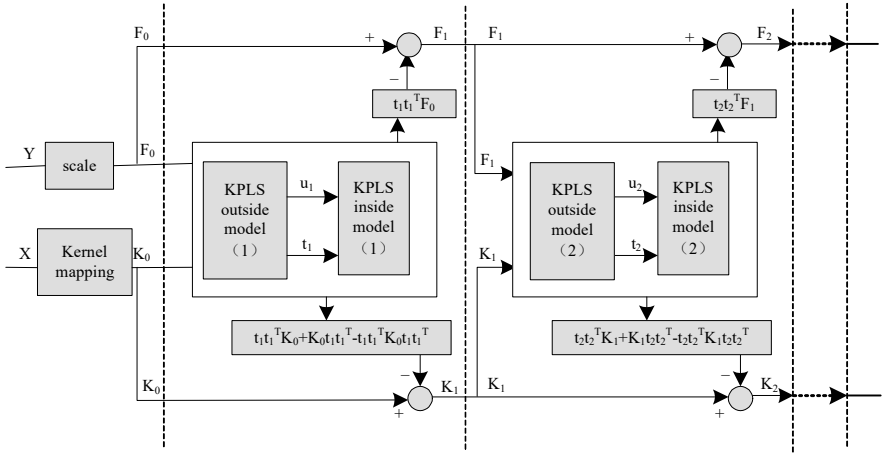

Figure 1-2: The structure of the KPLS algorithm

By extending a nonlinear item (using the "kernel trick") to the input matrix, the KPLS algorithm constructs a nonlinear model by using LVs in the kernel feature space.