

Wan Sieng Yeo<sup>1</sup> / Agus Saptoru<sup>1</sup> / Perumal Kumar<sup>1</sup>

# Development of Adaptive Soft Sensor Using Locally Weighted Kernel Partial Least Square Model

<sup>1</sup> Department of Chemical Engineering, Curtin University, Sarawak Campus, CDT 250, Miri 98009, Sarawak, Malaysia, E-mail: christineyeo@curtin.edu.my. <http://orcid.org/0000-0003-3248-3521>, <http://orcid.org/0000-0002-1734-4788>.

## Abstract:

Locally weighted partial least square (LW-PLS) model has been commonly used to develop adaptive soft sensors and process monitoring for numerous industries which include pharmaceutical, petrochemical, semiconductor, wastewater treatment system and biochemical. The advantages of LW-PLS model are its ability to deal with a large number of input variables, collinearity among the variables and outliers. Nevertheless, since most industrial processes are highly nonlinear, a traditional LW-PLS which is based on a linear model becomes incapable of handling nonlinear processes. Hence, an improved LW-PLS model is required to enhance the adaptive soft sensors in dealing with data nonlinearity. In this work, Kernel function which has nonlinear features was incorporated into LW-PLS model and this proposed model is named locally weighted kernel partial least square (LW-KPLS). Comparisons between LW-PLS and LW-KPLS models in terms of predictive performance and their computational loads were carried out by evaluating both models using data generated from a simulated plant. From the results, it is apparent that in terms of predictive performance LW-KPLS is superior compared to LW-PLS. However, it is found that computational load of LW-KPLS is higher than LW-PLS. After adapting ensemble method with LW-KPLS, computational loads of both models were found to be comparable. These indicate that LW-KPLS performs better than LW-PLS in nonlinear process applications. In addition, evaluation on localization parameter in both LW-PLS and LW-KPLS is also carried out.

**Keywords:** adaptive soft sensors, algorithm, just-in-time, kernel function, locally weighted partial least square, mathematical model

**DOI:** 10.1515/cppm-2017-0022

**Received:** May 5, 2017; **Accepted:** July 7, 2017

## 1 Introduction

Soft sensors have been actively used in various industrial applications to estimate product quality or other important variables when real time measurements are not available or online measurements are associated with long-time delay and/or difficult or expensive to be regularly carried out. According to the industrial questionnaire surveys on soft sensor applications in chemical process industries reported by [1] and [2], notice that soft sensors have been favorably applied in major unit operations such as distillation columns and reaction systems in numerous industrial process plants. These plants consist of semiconductor, steel-making, pharmaceutical, refinery and petrochemical processing industries. The survey's results are illustrated in Table 1 including the industrial applications of physical, data driven and hybrid (grey-box) models based soft sensors. Physical models (Phys), also called first principle models, are developed by using physicochemical knowledge of the process. Meanwhile, data-driven models include Multiple Linear Regression (MLR), Partial least square (PLS), Artificial Neural Network (ANN), Just-in-time (JIT) and other linear regressions (O.L.) are formulated from recorded historical operational data from the process plants. Grey-box models (Grey) are hybrid models between physical and data-driven models.

**Table 1:** Survey on the industrial applications of soft sensor [1, 2].

Process	Methods							Total
	Phys	MLR	PLS	O.L.	ANN	JIT	Grey	

Wan Sieng Yeo is the corresponding author.

© 2017 Walter de Gruyter GmbH, Berlin/Boston.

Distillation	20	256	41	6	0	5	3	331
Reaction	5	32	43	0	0	5	1	86
Polymerisation	0	4	8	0	3	0	5	20
Others	0	1	1	0	0	0	0	2
Total	25	293	93	6	3	10	9	439

It can also be seen in Table 1 that majority of industrial soft sensors are based on data-driven models. Among these data-driven models, MLR and PLS models are the most widely used ones. MLR models are popularly employed in industrial applications due to its simplicity in model development [2–5]. Nevertheless, these models are over-fitted and inaccurate when dealing with high-dimensional data and unable to cope with collinearity between process variables. Since process data from modern process plants exhibit high collinearity and involve large numbers of variables, PLS models are more preferable choices for these cases. Researchers, through comparative studies, have also indicated that overall PLS models perform better than MLR and PCR models in terms of predictive performances [2, 6].

In spite of the successful and wide application of soft sensors, the soft sensors involved some practical difficulties. Traditional soft sensors or so called the offline soft sensors such as PLS, MLR, and ANN were developed using plant historical data in offline modes before their implementations. The predictive performance of these kind of soft sensor will gradually deteriorate after a period of time due to changes in process characteristics and the state of plants, such as catalyst deactivation and sensor and process drifts due to fouling, equipment ageing, clogging and wear, and changes of raw materials [2, 7, 8]. In a survey that was carried out by industrial engineers, most of the engineers indicated that the major problem with soft sensors is the deterioration of the accuracy caused by changes in process conditions [1, 2].

Hence, from practical point of view, to deal with process changes and keep its good performance, a soft sensor should be updated routinely since the process characteristics change. Therefore, the development of adaptive soft sensors becomes essential. Algorithms used to develop adaptive soft sensors have been introduced and they can be found in the literature. Some of the algorithms have also been successfully executed in industrial applications such as Correlation-based JIT (CoJIT) and locally weighted kernel partial least square (LW-PLS) based adaptive soft sensor models [2, 6, 9, 10]. Nevertheless, CoJIT based adaptive soft sensors cannot deal with outliers and it is difficult to obtain its tuning parameter by trial-and-error [2, 7]. In comparison with CoJIT, LW-PLS can be suitable alternative approach as it can cope with outliers [6, 11] as well as collinearity and nonlinearity among process variables [2]. On the other hand, linear PLS regression in LW-PLS may not function well when processes behave highly nonlinear characteristics [12–14]. Thus, an improved algorithm for LW-PLS based adaptive soft sensors which is capable of dealing with highly nonlinear data is required.

To address this issue, several nonlinear versions of PLS such as quadratic PLS (QPLS), spline PLS (SPLS), neural network PLS (NNPLS) and kernel PLS (KPLS) models have been proposed. QPLS, SPLS and NNPLS which are internal nonlinear PLS (NPLS) models make use of nonlinear functions (e. g. simple polynomial transformation of observed data) to define relationship between latent variables (not directly observed variables) [15–17]. However, predefined form of quadratic function in QPLS algorithms has restricted its flexibility to develop nonlinear model [13, 18–20]. Although SPLS and NNPLS algorithms provide flexibility to capture nonlinearity relationships for variables, these methods may lead to local minima or obtain over-fitted model [18, 19, 21]. Different from the abovementioned internal NPLS models, KPLS model which was first introduced by [22] maps observed variables into a high-dimensional feature space using kernel function before linear PLS models are built in this new feature space. Compared to other NPLS models, Kernel-based techniques include KPLS model does not involve nonlinear optimization strategy [14] and KPLS model only needs linear algebra which is simple as a linear PLS regression [3, 23]. Moreover, different types of Kernels like polynomial Kernel, Gaussian Kernel, Sigmoid Kernel, exponential Kernel and Fourier Kernel functions [18] allow KPLS to handle different forms of nonlinearities [24]. Due to these advantages, KPLS had been used to capture nonlinear information from chemical processes by [14, 25, 26], and [23] for process monitoring as well as [27] for fault detection. In addition [3, 28], and [29] adopted KPLS to develop soft sensors for nonlinear process while [12] used modified KPLS for general modelling. On the other hand, Kernel functions was incorporated into Locally Weighted based algorithm to handle nonlinear problem by [30] for soft sensing and [31] for process monitoring.

It is to be noted that researchers have not paid enough attention to implement KPLS for soft sensor development compared to other applications such as process monitoring and fault detection. Meanwhile,

LW-PLS is unable to deal with highly nonlinear process data and an improved approach for LW-KPLS algorithm has not been found yet. To address these reach gaps, this paper proposed LW-KPLS to develop adaptive soft sensors in order to capture highly nonlinear relationships in process variables. Subsequently, better process sensing can be conducted and this means better process control and optimisation to assist process plants in reaching their operational excellence which conclusively leads to more sustainable and profitable operations.

## 2 Algorithms

In this section, formulation of LW-PLS model used to develop adaptive soft sensors is briefly reviewed and followed by description of KPLS model. Next, a brief overview on distance based similarity measure for LW-PLS and LW-KPLS algorithm is presented. Then, parameter optimization and measurement of quality prediction are also presented.

### 2.1 Locally weighted partial least square

Since LW regression reveals problematic when the number of input variables are more than the neighbours samples [32], LW-PLS has been found to be a suitable alternative when dealing with higher dimensionality of variables (a large number of input variables) and outliers [6, 11] as well as collinearity among the variables [2]. Up to date, LW-PLS based soft sensors have been applied to various industries such as biochemical, semiconductor, petrochemical, and pharmaceutical processing. This indicates that LW-PLS has become a popular algorithm in developing soft sensors [2, 4–6, 32, 33].

LW-PLS is a just-in-time based algorithm where a local PLS model is constructed based on similarity between a query (new data) and sample dataset [2, 34]. The formulation of LW-PLS algorithm which is taken from [6] is presented here. Input and output variables,  $x$  and  $y$  for  $n$  number of sample can be denoted as eqs (1) and (2) where  $M$  and  $L$  are numbers of  $x$  and  $y$ , respectively. These  $x$  and  $y$  are stored in a sample database. Similarity index,  $\omega_n$  between a query,  $X_q$  and the  $n^{\text{th}}$  historical data,  $X_k$  are calculated to construct a local PLS model when an output prediction,  $\hat{y}_q$  is required for  $X_q$ . A well explanation on this LW-PLS model can be found in [5, 33] and [35].

$$x_n = [x_{n1}, x_{n2}, \dots, x_{nM}]^T \quad (1)$$

$$y_n = [y_{n1}, y_{n2}, \dots, y_{nL}]^T \quad (2)$$

The predicted output,  $\hat{y}_q$  is calculated through:

1. Determine the number of latent variables  $K$  and set  $k = 1$ .
2. Calculate a similarity matrix  $\Omega$  using eq. (3).
3. Calculate  $X_k$ ,  $Y_k$ , and  $X_{q,k}$  using eqs (4)–(8) where  $1_N$  is a vector of ones.
4. Set  $\hat{y}_q = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_L]^T$ .
5. Derive the  $k^{\text{th}}$  latent variable of  $X_k$  using eq. (9) where  $w_k$  is the eigenvector of  $X_k^T \Omega Y_k Y_k^T \Omega X_k$ , which corresponds to the maximum eigenvalue, and it is derived as eq. (10).
6. Derive the  $k^{\text{th}}$  loading vector of  $X_k$  and the  $k^{\text{th}}$  regression coefficient vector using eqs (11) and (12) respectively.
7. Derive the  $k^{\text{th}}$  latent variable of  $X_q$  using eq. (13).
8. Replace  $\hat{y}_q$  with  $\hat{y}_q + t_{q,k} q_k$ .
9. If  $k = K$ , then finish prediction. Otherwise, set as eqs (14)–(16).
10. Set  $k = k + 1$  and go to Step 5.

$$\Omega = \text{dig} \{ \omega_1, \omega_2, \dots, \omega_N \} \quad (3)$$

$$X_k = X - 1_N \begin{bmatrix} \bar{X}_1 & \bar{X}_2 & \dots & \bar{X}_M \end{bmatrix} \quad (4)$$

$$Y_k = Y - 1_N \begin{bmatrix} \bar{Y}_1 & \bar{Y}_2 & \dots & \bar{Y}_L \end{bmatrix} \quad (5)$$

$$X_{q,k} = X_q - 1_N \left[ \bar{X}_1, \bar{X}_2, \dots, \bar{X}_M \right]^T \quad (6)$$

$$\bar{X}_m = \frac{\sum_{n=1}^N \omega_n X_{nm}}{\sum_{n=1}^N \omega_n} \quad (7)$$

$$\bar{Y}_l = \frac{\sum_{n=1}^N \omega_n Y_{nl}}{\sum_{n=1}^N \omega_n} \quad (8)$$

$$t_k = X_k w_k \quad (9)$$

$$w_k = \frac{X_k^T \Omega Y_k}{\|X_k^T \Omega Y_k\|} \quad (10)$$

$$p_k = \frac{X_k^T \Omega t_k}{t_k^T \Omega t_k} \quad (11)$$

$$q_k = \frac{Y_k^T \Omega t_k}{t_k^T \Omega t_k} \quad (12)$$

$$t_{q,k} = X_{q,k}^T w_k \quad (13)$$

$$X_{k+1} = X_k - t_k p_k^T \quad (14)$$

$$Y_{k+1} = Y_k - t_k q_k^T \quad (15)$$

$$X_{q,k+1} = X_{q,k} - t_{q,k} p_k \quad (16)$$

## 2.2 Kernel partial least square

Kernel methods have been actively applied to solve nonlinear problem due to better performance in most real-world applications [26]. Hence, Kernel functions were incorporated into PLS model to enhance the predictive performance of PLS in dealing with nonlinear data. KPLS first maps original observed data into a high dimensional feature space and then linear PLS models are constructed in the feature space. Initially, it is assumed that there is a nonlinear transformation of independent variables,  $x_i, i = 1, 2, \dots, n$  into feature space  $F$  and it can be represented using eq. (17) where it is also assumed that  $\sum_{i=1}^n \varphi(x_i) = 0$ .  $\varphi(x_i)$  is a nonlinear mapping function that projects the input vectors from the original space to  $F$ . On the other hand, the dimensionality of the feature space  $F$  is arbitrarily large, and can be infinite. The order of matrix  $\varphi(x)$  is  $n \times S$  where  $i$ -th row is the vector  $\varphi(x_i)$  in the  $S$ -dimensional feature space  $F$ . By means of the introduction of the kernel trick,  $\varphi(x_i)^T \varphi(x_j) = K(x_i, x_j)$ , one can prevent both performing explicit nonlinear mapping and computing dot products in the feature space. Steps for KPLS model from [12] are shown as follows:

Let  $K$  is the number of latent variables ( $i = 1, 2, \dots, K$ ) and the below steps are repeated.

1. Initialise, set  $K_i = K$ ,  $Y_i = Y$ , set  $u_i$  equal to any column of  $Y_i$ .
2. Compute the score vector using eq. (18).
3. Compute the loading vector of  $Y_i$  using eq. (19).
4. Compute the score vector of  $Y_i$  using eq. (20).
5. If  $u_i$  converges, then go to Step 6, else return to Step 2.
6. Calculate the residual of  $\varphi(X)$  and  $Y$  using eqs (21) and (22).
7. Then,  $i = i + 1$ , and go to Step 2.

$$\phi : x_i \in R^n \rightarrow \varphi(x_i) \in F \quad (17)$$

$$\phi(X) : t_i = \frac{Ku_i}{\sqrt{u_i^T K_i u_i}} \quad (18)$$

$$q_i = \frac{Y_i t_i}{\|t_i^T t_i\|} \quad (19)$$

$$u_i = \frac{Y_i q_i}{q_i^T q_i} \quad (20)$$

$$K_{i+1} = (I - t_i t_i^T / t_i^T t_i) K_i (I - t_i t_i^T / t_i^T t_i) \quad (21)$$

$$Y_{i+1} = (I - t_i t_i^T / t_i^T t_i) Y_i \quad (22)$$

The regression coefficient,  $b$  in KPLS is determined from eq. (23) after all the latent variables are extracted. In eq. (23),  $T = [t_1, t_2, \dots, t_H]$  and  $U = [u_1, u_2, \dots, u_H]$  are the score matrix. Hence, when the number of test data is  $n_t$ , the prediction on training data and test data can be made as eqs (24) and (25) respectively: In both of these equations,  $\varphi_{test}$  is the matrix of the mapped test data and  $K_{test}$  is the  $n_t \times n$  test kernel matrix whose elements are  $K_{test}(i, j) = K(x_i, x_j)$ , where  $x_i$  is the  $i$ -th test vector and  $x_j$  is the  $j$ -th training vector.

$$b = \phi^T U (T^T K U)^{-1} T^T Y \quad (23)$$

$$\hat{Y}_{train} = \phi b = K U (T^T K U)^{-1} T^T Y \quad (24)$$

$$\hat{Y}_{test} = \phi_{test} b = K_{test} U (T^T K U)^{-1} T^T Y \quad (25)$$

Mean centering of the data should be done before performing KPLS [36]. The kernel matrices  $K$  and  $K_{test}$  are substituted with  $\tilde{K}$  and  $\tilde{K}_{test}$ . They can be represented by eqs (26) and (27) where  $1$  is an  $n$ -dimensional identity matrix,  $1_n$  and  $1_{n_t}$  are vectors whose elements are all ones, with length  $n$  and  $n_t$ , respectively.

$$\tilde{K} = \left( I - \frac{1}{n} 1_n 1_n^T \right) K \left( I - \frac{1}{n} 1_n 1_n^T \right) \quad (26)$$

$$\tilde{K}_{test} = \left( K_{test} - \frac{1}{n} 1_{n_t} 1_n^T K \right) \left( I - \frac{1}{n} 1_n 1_n^T \right) \quad (27)$$

### 2.3 Similarity measure

Prediction quality of LW-PLS is significantly affected by definition of similarity [6]. Thus, the similarity has to be well designated. In this study, Euclidean distance, which is a widely used similarity measure is employed since it is easier to implement [3, 7, 37, 38]. The Euclidean distance based similarity index,  $\omega_n$  is calculated based on distance between  $x_q$  and  $x_n$ . The similarity index from [39] is shown in the eqs (28) and (29) where  $\phi$  is a localization parameter,  $\sigma_d$  is standard deviation of  $d_n$  ( $n = 1, 2, \dots, N$ ). The similarity index reduces dramatically when  $\phi$  is a small value and slowly when  $\phi$  is a big value. In order to tackle nonlinear relationship between input and output variables, small value of  $\phi$  is required. If the value of  $\phi$  is too small, sensitivity of LW-PLS towards noise is then increased [4]. On the other hand, predictive performance of LW-PLS is same as PLS when  $\phi$  equals to infinite [32, 34]. Hence, proper setting of  $\phi$  is needed to ensure the prediction accuracy of LW-PLS is superior to PLS.

$$\omega_n = \exp\left(-\frac{d_n}{\phi\sigma_n}\right) \quad (28)$$

$$d_n = \sqrt{(x_n - x_q)^T (x_n - x_q)} \quad (29)$$

### 2.4 Locally weighted kernel partial least square

Although LW-PLS can achieve good performance in industrial applications, it still may not fulfill the prediction accuracy requirement for highly nonlinear processes. In this study, LW-KPLS is introduced to handle this issue. First, original variables are mapped into high dimensional space using Kernel function and then locally weight models are built using the same procedure as LW-PLS. Similar to the procedure of LW-PLS, before constructing locally weighted model, this new high dimension feature variable vectors,  $\varphi(x)$  should also be weighted-mean centered. It is assumed that simplicity is fulfilled and then LW-KPLS can be formulated as below.

Since original variables are mapped into a kernel-defined feature space, correlations between mapped input variables and predicted output variables are required to be identified. This can be done by performing dual KPLS discrimination to measure the projection direction in the kernel-defined feature space that contains the maximum amount of variance in the data. This discrimination is done by obtaining a dual representation of a scaled version of projection direction,  $B$  that moved a Kernel matrix,  $V$ . According to [40], it can be calculated by eq. (25). The detail explanation on eq. (30) and matlab coding for eq. (30) can be found in [40].

The predicted output,  $\hat{y}_q$  is calculated through the following procedure:

1. Determine Kernel matrices for input variables,  $V$  and query,  $V_q$  where they are mapped into higher dimension feature space using selected Kernel function.
2. Conduct mean centering on the  $V$  and  $V_q$  using eqs (26) and (27).
3. Perform dual kernel partial least square discrimination to obtain  $B$  using eq. (30).
4. Compute re-scaled query, and input variable matrices,  $V_q$ , and  $V$  using eqs (31) and (32).
5. Determine the number of latent variables  $K$  and set  $k = 1$ .
6. Calculate a similarity matrix  $\Omega$  using eqs (3), (28) and (29).
7. Calculate  $X_k$ ,  $Y_k$ , and  $X_{q,k}$  using eqs (4)–(8).
8. Set  $\hat{y}_q = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_L]^T$ .
9. Derive the  $k^{\text{th}}$  latent variable of  $X_k$  using eqs (9) and (10).
10. Derive the  $k^{\text{th}}$  loading vector of  $X_k$  and the  $k^{\text{th}}$  regression coefficient vector using eqs (11) and (12).
11. Derive the  $k^{\text{th}}$  latent variable of  $X_q$  using eq. (13).
12. Replace  $\hat{y}_q$  with  $\hat{y}_q + t_{q,j}q_j$  where  $t_{q,k}$  is the  $k^{\text{th}}$  latent variable of  $X_q$ .
13. If  $k = K$ , then finish prediction. Otherwise, set eqs (14)–(16).

14. Set  $k = k + 1$  and go to Step 9.

$$B = YY' V\beta \text{ with the normalisation, } \beta = \frac{\beta}{\|\beta\|} \quad (30)$$

$$X_q = V_q B \quad (31)$$

$$X = VB \quad (32)$$

## 2.5 Parameter optimization and measurement of quality prediction

Optimized parameter is required to be tuned in order to find a well performing model. From above Subsections Section 2.1 and Section 2.4, it can be noticed that both LW-PLS and LW-KPLS involve some similar optimized parameters which are localization parameter,  $\phi$ , number of latent variables,  $K$ , and number of sample,  $N$ . Besides, LW-KPLS needs another optimized parameter which is Kernel parameter.

In this study,  $K$ ,  $N$ , and  $\phi$  are fixed to proper values while Kernel parameter is set by trial and error experiments. Since  $N$  is not a sensitive parameter for both algorithms, it is specified as a same proper value in case study. Normally, the first few latent variables which are so called principal components of PLS based algorithms can sufficiently describe the main feature of data [30]. Furthermore, same value of  $K$  is applied in case study in order to compare both algorithms fairly. Meanwhile, considering different values of  $\phi$  in both algorithms do not change the results as  $\phi$  is only useful when choosing related samples for query sample in both algorithms [30]. Hence,  $\phi$  is also fixed as same value for both algorithms in case study. On the other hand, Kernel parameter which is a very sensitive parameter for both algorithms to be tuned in order to obtain smallest error of dataset which include training and test set. Table 2 summarises all the parameters used in both algorithms.

**Table 2:** Parameters used in the algorithms.

Algorithm	Used parameters
LW-PLS	$\phi, K, N$
LW-KPLS	$\phi, K, N, \text{Kernel parameter}$

In order to measure and compare prediction accuracy of the models, an over-fitting model, which is called as “the error of the test set is not the smallest” is used. There is a possible network where the error of the training set is smallest, however the error of test set is large [41]. This form of network is unsteady when it is utilized to predict an unknown samples. To prevent these situations, a new evaluation criterion of network, the error of approximation,  $E_a$  is used. The description of this criterion that is taken from [42] is given in eqs (33) and (34) where  $RMSE_1$  and  $RMSE_2$  are root mean square errors of the training and the test sets.  $RMSE_1$  and  $RMSE_2$  are used to evaluate the ability of a model to fit data and predictive power [28]. Equation of root mean square error (RMSE) is shown in eq. (34) where  $N$ ,  $N_1$  and  $N_2$  are the total number of samples, the number of training and test sets, respectively. Meanwhile,  $y_i$  and  $\hat{y}_i$  are the real and estimated values of output. According to [41] and [42], smaller  $E_a$  indicates that more models are close to the real nature of the dataset. Hence, the effects of both training and test sets are examined using this evaluation criterion [41].

$$E_a = \left(\frac{N_1}{N}\right) RMSE_1 + \left(\frac{N_2}{N}\right) RMSE_2 + |RMSE_1 - RMSE_2| \quad (33)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}} \quad (34)$$

### 3 Methodology

A case study is used to demonstrate the performance of LW-KPLS and the obtained results are compared with LW-PLS. In this section, splitting data and setting of parameters that are utilised for both algorithms in this case study are presented. Then, the details of this case study is briefed and followed by explanation on the configuration of computer.

#### 3.1 Splitting data and parameters setting

A total of 10,000 dataset is split into training and test sets. Hence, for the case study, 75 % of the dataset are selected as training data and another 25 % of dataset are used as test data. Hence, in the case study,  $N$ ,  $N_1$  and  $N_2$  are set as 10,000, 7500 and 2500 respectively. Meanwhile,  $K$  and  $\phi$  for both algorithms are adjusted as 1 and 0.001 respectively in the case study. Table 3 illustrates the values of these used parameters for both algorithms. Besides, polynomial kernel function  $k(X_i, X_j) = -(bX_i'X_j + b)^b$  where  $b$  is Kernel parameter is chosen to be incorporated into LW-KPLS in the case study.

**Table 3:** Values of used parameters for both algorithms.

Used parameters	Values
$N$	10,000
$N_1$	7,500
$N_2$	2,500
$K$	1
$\phi$	0.1

#### 3.2 Case study

A case study of nonlinear processes are used to evaluate predictive performance of LW-KPLS and LW-PLS algorithms. This case study is adopted from [43] which is simulated dataset from a biological network that arises in eukaryotic cell cycle regulation. Process description of this case study is presented in the below subsections.

##### 3.2.1 Case study: Eukaryotic cell cycle regulation

This process is a biochemical reaction that involves cyclin-dependent kinase and their associated proteins, which are included in cell cycle to control frog egg development. There are three key species in this biochemical reaction which are free cyclin, the M-phase promoting factor (MPF) and other regulatory enzymes. MPF is heterodimer which is formed by the proteins Cdc2 and cyclin. It controls the transition from interphase to mitosis of the cell cycle. In this study, it is assumed that this Cdc2-cyclin regulatory system operates as a spontaneous oscillator in early embryos. The detailed description of this process is given in [43].

In this study, the predictive output variables used to estimate product quality is a dimensionless concentration of total cyclin which is denoted as  $v$ . This output variable is measured using a dimensionless concentration of active MPF,  $u$ . Table 4 shows input and output variables for the case study.

**Table 4:** Input and output variables for the case study.

Output variables	Variable description
$v$	Dimensionless concentration of total cyclin
Input variables	Variable description
$u$	Dimensionless concentration of active MPF

The configuration of the computer is stated as below:

Operating System: Window 10 (64 bit); CPU: 2.20 GHz Intel core M3-6Y30 (same performance as i-5 6200U) CPU processor, 4.0 GB of RAM, 128GB SSD storage. The version of Matlab is 2015a.



## 4 Results and discussion

In this section, application results of LW-PLS and LW-KPLS to the case study are reported. For this case study, predictive performance of both algorithms are compared and discussed. Meanwhile, evaluation on localization parameter that is used in LW-PLS and LW-KPLS models is also carried out.

### 4.1 Case study

In this case study, the prediction results of LW-PLS and LW-KPLS for the training and test data are summarised in Table 5. Notice that LW-PLS obtained same RMSE for both training and test data where  $RMSE_1$  and  $RMSE_2$  are 0.0713 with  $E_a$ , central processing unit (CPU) running time for both training and test data,  $CPU_1$  and  $CPU_2$  are 0.074, 8 sec and 3 sec respectively. LW-KPLS shows better predictive ability for both training and test data where  $RMSE_1$ ,  $RMSE_2$  and  $E_a$  are 0.0692, 0.0697, and 0.0698 respectively. However, its  $CPU_1$  and  $CPU_2$  are 513 sec and 142 sec respectively. With the help of Kernel function, the  $RMSE_1$ ,  $RMSE_2$  and  $E_a$  of LW-KPLS are improved by about 2.94 %, 2.31 % and 5.67 % respectively, in comparison with LW-PLS. These results show clearly that LW-KPLS performs better in this nonlinear process. Figure 1 and Figure 2 present prediction results for training and test data of case study using LW-PLS and LW-KPLS. However, computational load for LW-KPLS is much higher than LW-PLS. The strategic to overcome this problem is illustrated in Subsection Section 4.2.

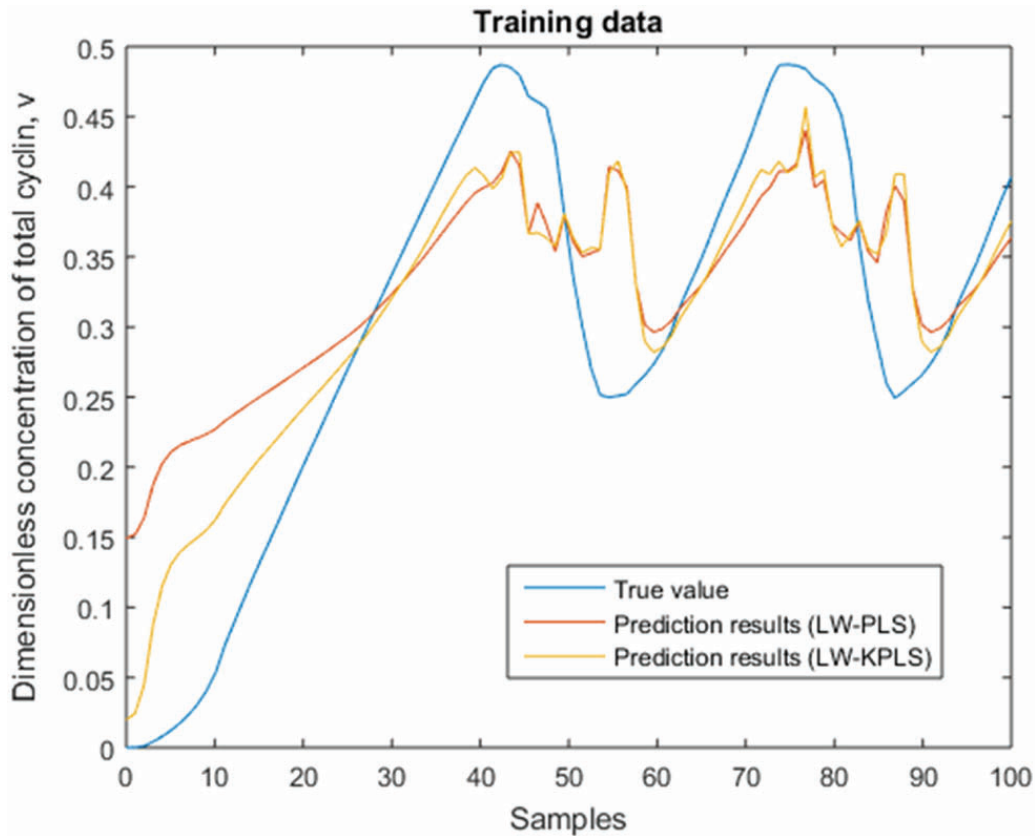


Figure 1: Prediction results for training data of the case study using LW-PLS and LW-KPLS.

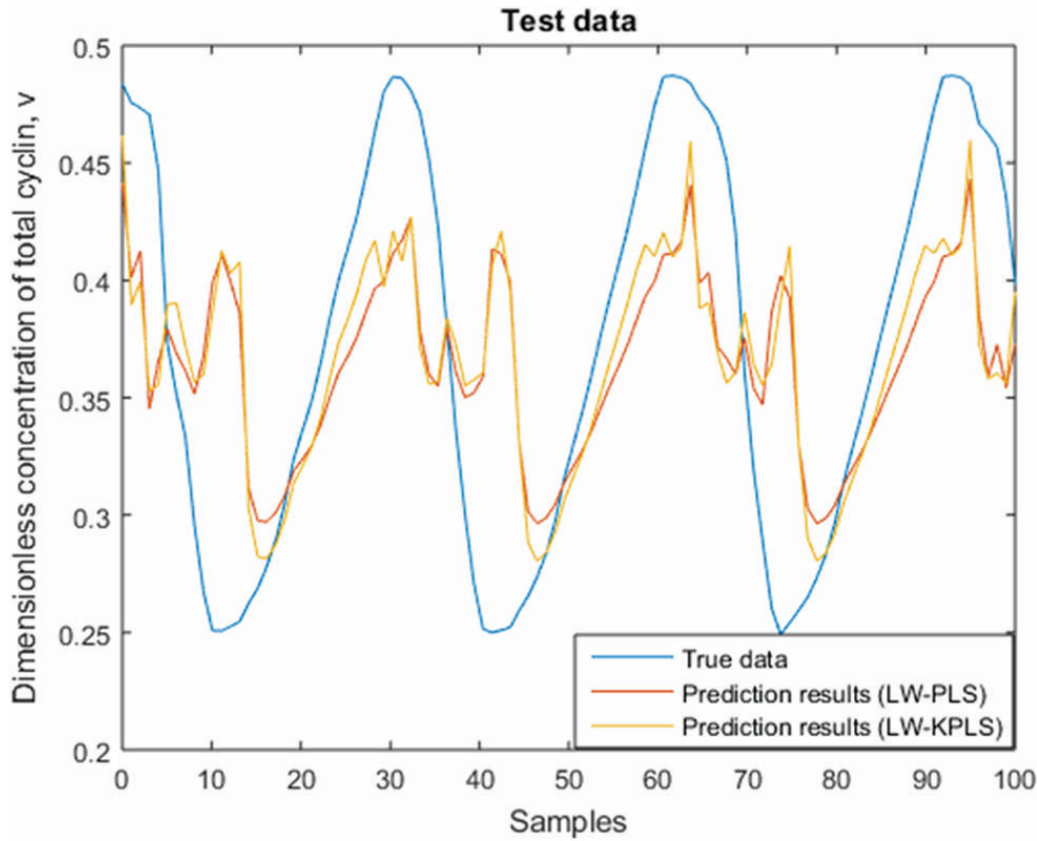


Figure 2: Prediction results for testing data of the case study using LW-PLS and LW-KPLS.

Table 5: Summary of prediction results of LW-PLS based and LW-KPLS based soft sensors for the case study.

Models	LW-PLS	LW-KPLS	%
$RMSE_1$	0.0713	0.0692 (b=51)	2.94
$RMSE_2$	0.0713	0.0697 (b=54)	2.31
$E_a$	0.0740	0.0698	5.67
$CPU_1$ (sec)	8	513	>100
$CPU_2$ (sec)	3	142	>100

### 4.2 Computational loads

Ensemble method is applied to LW-KPLS in order to reduce its computational load. Both training and test data are separated into several same dimension data sets and these sets are run using LW-KPLS simultaneously. Then, the obtained CPU times and prediction performance of LW-KPLS in different number of sets for the case study is shown in Table 6. From Table 6, notice that  $CPU_1$  is higher than  $CPU_2$  as  $N_1$  is much higher than  $N_2$  (Please refer to Table 3). It can be seen from Table 6 that  $CPU_1$  is reduced from 513 sec to 48 sec while  $CPU_2$  is decreased from 142 sec to 10 sec after divided the dataset into 4 sets. Only 4 sets of results are presented since no significant improvement for the further division of dataset. Meanwhile,  $RMSE_1$ ,  $RMSE_2$ , and  $E_a$  stated in Table 6 are plotted in Figure 3. Figure 3 shows the error lines for training and test data of the case study obtained by LW-PLS and different number of sets for LW-KPLS. As can be seen in Figure 3,  $RMSE_2$  and  $E_a$  for LW-KPLS are increasing while  $RMSE_1$  is decreasing when the number of set is getting higher. Compared to LW-PLS,  $RMSE_1$ ,  $RMSE_2$  and  $E_a$  for LW-KPLS are still lower. From the above results, LW-KPLS performs better than LW-PLS in term of prediction performance for nonlinear processes. Although LW-KPLS using ensemble method that divided data into 4 sets of data has slightly higher computational loads than LW-PLS, it is still comparable.

Automatically generated rough PDF by ProofCheck from River Valley Technologies Ltd

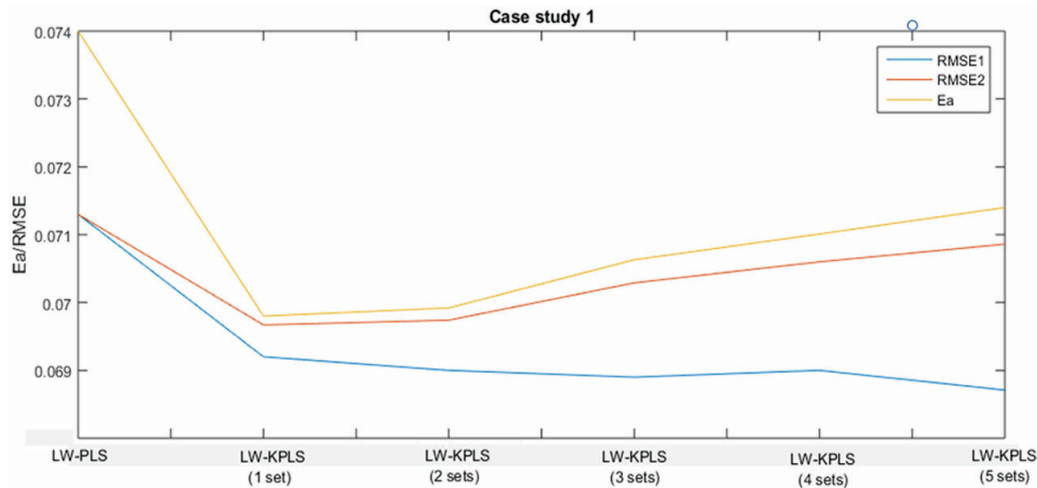


Figure 3: The error lines for training and test data of the case study obtained by LW-PLS and different sets of LW-KPLS.

Table 6: CPU time and prediction performance of LW-KPLS in different sets for the case study.

Number of set	1	2	3	4
$RMSE_1$	0.0692 (b=51)	0.0690 (b=53)	0.0689 (b=49)	0.0690 (b=44)
$RMSE_2$	0.0697 (b=54)	0.0697 (b=53)	0.0703 (b=59)	0.0706 (b=51)
$E_a$	0.0698	0.0699	0.0706	0.0710
$CPU_1$ (sec)	513	135	64	48
$CPU_2$ (sec)	142	34	19	10

### 4.3 Evaluation on localization parameter

The localization parameter,  $\phi$  in eq. (23) is a tuning parameter that is obtained to minimise the RMSE of data set in the model development [35]. It is used to determine similarity index,  $\omega_n$  for LW-PLS and LW-KPLS models.  $\omega_n$  reduces in an exponential manner and closes to zero when the distance from query increases. Moreover,  $\omega_n$  reduces steeply when  $\phi$  becomes small while  $\omega_n$  decreases gradually when  $\phi$  is large.

On the other hand, accordingly to [4],  $\phi$  is also used to identify the nonlinearity of the soft sensors; the soft sensor can deal with process nonlinearity when  $\phi$  is small and the soft sensor becomes overfitting when  $\phi$  is too small. As mentioned earlier in Subsection Section 3.1, this parameter which is to be used in this work is set as 0.1.

In this subsection, different values of  $\phi$ , which are 0.0005, 0.001, 0.1, 0.15 and 0.2 are examined. These  $\phi$  values are chosen to be observed since nonlinear process requires small value of  $\phi$  [5]. Both LW-PLS and LW-KPLS are applied to the case study using these different values of  $\phi$ . Then, the  $E_a$  values for LW-PLS and LW-KPLS are determined and the results are illustrated in Table 7.

Table 7:  $E_a$  values for LW-PLS and LW-KPLS in different sets for the case study using different values of  $\phi$ .

$\phi$	LW-PLS	LW-KPLS			
		Number of set			
		1	2	3	4
0.0005	0.1653	0.1613	0.1064	0.1474	0.1291
0.001	0.1368	0.0775	0.0956	0.1066	0.1356
0.1	0.0740	0.0698	0.0699	0.0706	0.0710
0.15	0.0736	0.0707	0.0708	0.0712	0.0714
0.2	0.0751	0.0720	0.0719	0.0720	0.0722

As shown in Table 7, all of the  $E_a$  values for LW-KPLS in different sets are lower than LW-PLS. This indicates that LW-KPLS with the  $\phi$  values of 0.005, 0.001, 0.1, 0.15 and 0.2 performs better than LW-PLS in dealing with nonlinear processes. In comparison of different values of  $\phi$ , it is found that LW-KPLS with  $\phi$  value of 0.1 ob-

tained the lowest values of  $E_a$  in each set. Hence, based on the fine-tuning of  $\phi$ , it was found that the optimum value is 0.1.

## 5 Conclusions

In this research, a novel algorithm, namely LW-KPLS which utilise Euclidean distance measure was used to develop adaptive soft sensors in order to solve nonlinear problem in chemical process plants. By using a case study of nonlinear processes, estimation accuracy of LW-PLS and LW-KPLS were demonstrated. Results demonstrated that the prediction performance of the proposed approach, LW-KPLS is superior to LW-PLS when dealing with nonlinear processes. However, this approach assumes that the process data are complete. Hence, future work can focus on addressing the missing data problem.

New York, USA

## Funding

This research is co-funded by Fundamental Research Grant Scheme (FRGS/2/2014/TK05/CURTIN/02/1), Ministry of Education, Malaysia and Curtin University Malaysia under Staff Study Support.

## References

- [1] Kano M, Ogawa M. The state of the art in chemical process control in Japan: good practice and questionnaire survey. *J Process Control*. 2010;20:969–982.
- [2] Kano M, Koichi F. Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications. *J Chem Eng Japan*. 2013;46:1–17.
- [3] Jin H, Chen X, Yang J, Wu L. Adaptive soft sensor modeling framework based on just-in-time learning and kernel partial least squares regression for nonlinear multiphase batch processes. *Comput Chem Eng*. 2014;71:77–93.
- [4] Kim S, Kano M, Hasebe S, Takinami A, Seki T. Long-term industrial applications of inferential control based on just-in-time soft-sensors: economical impact and challenges. *Ind Eng Chem Res*. 2013;52:12346–12356.
- [5] Kim S, Okajima R, Kano M, Hasebe S. Development of soft-sensor using locally weighted PLS with adaptive similarity measure. *Chemom Intell Lab Syst*. 2013;124:43–49.
- [6] Hazama K, Kano M. Covariance-based locally weighted partial least squares for high-performance adaptive modeling. *Chemom Intell Lab Syst*. 2015;146:55–62.
- [7] Saptoro A.. State of the art in the development of adaptive soft sensors based on just-in-time models. *Procedia Chemistry Elsevier*, 2014;226–234.
- [8] Kadlec P, Grbić R, Gabrys B. Review of adaptation mechanisms for data-driven soft sensors. *Comput Chem Eng*. 2011;35:1–24.
- [9] Jin H, Chen X, Yang J, Wang L, Wu L. Online local learning based adaptive soft sensor and its application to an industrial fed-batch chlortetracycline fermentation process. *Chemom Intell Lab Syst*. 2015;143:58–78.
- [10] Jin H, Chen X, Yang J, Zhang H, Wang L, Wu L. Multi-model adaptive soft sensor modeling method using local learning and online support vector regression for nonlinear time-variant batch processes. *Chem Eng Sci*. 2015;131:282–303.
- [11] Kaneko H, Funatsu K. Adaptive database management based on the database monitoring index for long-term use of adaptive soft sensors. *Chemom Intell Lab Syst*. 2015;146:179–185.
- [12] Gao Y, Kong X, Hu C, Zhang Z, Li H, Hou LA. Multivariate data modeling using modified kernel partial least squares. *Chem Eng Res Des*. 2015;94:466–474.
- [13] Shan P, Peng S, Tang L, Yang C, Zhao Y, Xie Q, et al. A nonlinear partial least squares with slice transform based piecewise linear inner relation. *Chemom Intell Lab Syst*. 2015;143:97–110.
- [14] Hu Y, Ma H, Shi H. Enhanced batch process monitoring using just-in-time-learning based kernel partial least squares. *Chemom Intell Lab Syst*. 2013;123:15–27.
- [15] Wang L, Yang D, Fang C, Chen Z, Lesniewski PJ, Mallavarapu M, et al. Application of neural networks with novel independent component analysis methodologies to a Prussian blue modified glassy carbon electrode array. *Talanta*. 2015;131:395–403.
- [16] Rosipal R. Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques In: Lodhi H, Yamanishi Y, editors. *Nonlinear partial least squares: an overview*. Hershey, PA: IGI Global, 2010: 169-189.
- [17] Frank IE. A nonlinear PLS model. *Chemom Intell Lab Syst*. 1990;8:109–119.
- [18] Wang Y, Cao H, Zhou Y, Zhang Y. Nonlinear partial least squares regressions for spectral quantitative analysis. *Chemom Intell Lab Syst*. 2015;148:32–50.
- [19] Shan P, Peng S, Bi Y, Tang L, Yang C, Xie Q, et al. Partial least squares–slice transform hybrid model for nonlinear calibration. *Chemom Intell Lab Syst*. 2014;138:72–83.
- [20] Zhou YP, Jiang JH, Lin WQ, Xu L, Wu HL, Shen GL, et al. Artificial neural network-based transformation for nonlinear partial least-square regression with application to QSAR studies. *Talanta*. 2007;71:848–853.

- [21] Wold S. Nonlinear partial least squares modelling II. Spline inner relation. *Chemom Intell Lab Syst.* 1992;14:71–84.
- [22] Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. *J Machine Learn Res.* 2002;2:97–123.
- [23] Zhang Y, Zhang Y. Complex process monitoring using modified partial least squares method of independent component regression. *Chemom Intell Lab Syst.* 2009;98:143–148.
- [24] Hu Y, Wang L, Ma H, Shi H. Online nonlinear process monitoring using kernel partial least squares [J]. *Ciesc J.* 2011;9:025.
- [25] Zhang Y, Li S, Hu Z, Song C. Dynamical process monitoring using dynamical hierarchical kernel partial least squares. *Chemom Intell Lab Syst.* 2012;118:150–158.
- [26] Zhang Y, Hu Z. On-line batch process monitoring using hierarchical kernel partial least squares. *Chem Eng Res Des.* 2011;89:2078–2084.
- [27] Jia Q, Zhang Y. Quality-related fault detection approach based on dynamic kernel partial least squares. *Chem Eng Res Des.* 2016;106:242–252.
- [28] Zhang Y, Teng Y, Zhang Y. Complex process quality prediction using modified kernel partial least squares. *Chem Eng Sci.* 2010;65:2153–2158.
- [29] García-Reiriz A, Damiani PC, Olivieri AC. Residual bilinearization combined with kernel-unfolded partial least-squares: a new technique for processing non-linear second-order data achieving the second-order advantage. *Chemom Intell Lab Syst.* 2010;100:127–135.
- [30] Yuan X, Ge Z, Song Z. Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes. *Ind Eng Chem Res.* 2014;53:13736–13749.
- [31] Jiang Q, Yan X. Weighted kernel principal component analysis based on probability density estimation and moving window and its application in nonlinear chemical process monitoring. *Chemom Intell Lab Syst.* 2013;127:121–131.
- [32] Toshiya H, Kano M. Adaptive virtual metrology design for semiconductor dry etching process through locally weighted partial least squares. *IEEE Trans Semicond Manuf.* 2015;28:137–144.
- [33] Nakagawa H, Tajima T, Kano M, Kim S, Hasebe S, Suzuki T, et al. Evaluation of infrared-reflection absorption spectroscopy measurement and locally weighted partial least-squares for rapid analysis of residual drug substances in cleaning processes. *Anal Chem.* 2012;84:3820–3826.
- [34] Kim S, Kano M, Nakagawa H, Hasebe S. Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection. *Int J Pharm.* 2011;421:269–274.
- [35] Nakagawa H, Kano M, Hasebe S, Miyano T, Watanabe T, Wakiyama N. Verification of model development technique for NIR-based real-time monitoring of ingredient concentration during blending. *Int J Pharm.* 2014;471:264–275.
- [36] Schölkopf B, Smola A, Müller KR. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* 1998;10:1299–1319.
- [37] Zhang X, Li Y, Kano M. Quality prediction in complex batch processes with just-in-time learning model based on non-gaussian dissimilarity measure. *Ind Eng Chem Res.* 2015;54:7694–7705.
- [38] Xie L, Zeng J, Gao C. Novel just-in-time learning-based soft sensor utilizing non-Gaussian information. *IEEE Trans Control Syst Technol.* 2014;22:360–368.
- [39] Ma M, Khatibisepehr S, Huang B. A Bayesian framework for real-time identification of locally weighted partial least squares. *AIChE J.* 2015;61:518–529.
- [40] Shawe-Taylor J, Cristianini N. *Kernel methods for pattern analysis.* New York, USA: Cambridge University Press, 2004.
- [41] Dou Y, Sun Y, Ren Y, Ren Y. Artificial neural network for simultaneous determination of two components of compound paracetamol and diphenhydramine hydrochloride powder on NIR spectroscopy. *Anal Chim Acta.* 2005;528:55–61.
- [42] Saptoro A, Hari BV, Moses OT. Design of feed-forward neural networks architecture for coal elemental composition prediction. *International conference on modeling and simulation, Kuala Lumpur, Malaysia;* 2006;3–5.
- [43] Novak B, Tyson JJ. Modeling the cell division cycle: M-phase trigger, oscillations, and size control. *J Theor Biol.* 1993;165:101–134.