Soft-Sensing Estimation of Optical Density for PHA Production Using Multilayer Perceptron Neural Network

Nor Hana Mamat\textsuperscript{1,4*}, Samsul Bahari Mohd Noor\textsuperscript{1}, Azura Che Soh\textsuperscript{1}, Farah Saleena Taip\textsuperscript{2}, Ahmad Hazri Ab. Rashid\textsuperscript{3}, Nur Liyana Jufika Ahmad\textsuperscript{3}, Ishak Mohd Yusof\textsuperscript{3}, Adida Zuraida Mohamad\textsuperscript{3}

\textsuperscript{1}Control System and Signal Processing Research Centre, Department of Electrical & Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor.

\textsuperscript{2}Food and Process Department, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor.

\textsuperscript{3}Industrial Biotechnology Research Centre, SIRIM Berhad, Persiaran Dato’ Menteri, Seksyen 2, 40700 Shah Alam, Selangor.

\textsuperscript{4}Faculty of Electrical and Automation Engineering Technology, TATI University College, Teluk Kalong, 24000 Kemaman, Terengganu.

Abstract. Biomass concentration is an important indicator of production rate in polyhydroxyalkanoates (PHA) fermentation process. In current practice, measurement of biomass concentration is done off-line by laboratory analysis that is unsuitable for online process monitoring and control. Soft-sensor is often used as an alternative that provides an estimate of hard to measure parameters from easy to measure process data. However, most of these studies use simulated data or data generated from mathematical model that was developed without full consideration of process and measurement uncertainty. In this study, a soft-sensor is developed from real production data for PHA fermentation in pilot-scale bioreactor with the appropriate data pre-processing techniques applied to process data that was obtained from this system. Multilayer perceptron (MLP) neural network is used to estimate biomass concentration using secondary process parameters such as dissolved oxygen (DO), temperature, pH and agitation speed as inputs. Different models are developed based on different batches of production data and various network architecture in order to study the appropriate integration of process data and network topology that gives the best model accuracy. Results indicate that the biomass soft-sensor developed using MLP-ANN provides a better estimate of biomass in comparison to radial basis function (RBF) neural network and support vector regression (SVR) methods. The developed soft-sensor can be further used in monitoring and control of production output.
1.0 Introduction

In bioprocess the output variable that are related to the quality of product is usually a biological parameter rather than physical or chemical variables. Current online sensors to measure biological parameters are either not available, high operational and maintenance cost or difficult to maintain by end user [1]. In bioplastic production of polyhydroxyalkanoates (PHA), optical density (OD) measurement is often used as indicator for biomass which is the fermentation process output quality [2]. Measurement of OD, alternatively known as turbidity is among the simplest and most frequently used biomass monitoring technology and has been used in many microbial, fungi and mammalian bioprocesses [1], [3]. OD values are commonly obtained through offline analysis where the product is sampled, and time-consuming laboratory analysis is performed. It is not suitable for online monitoring and control of the fermentation process.

A review on biomass estimation methods [4] suggested the use of optical density measurement and software sensor method using the commonly measured bioreactor’s online process variables. This method is an economical means for biomass estimation as it uses existing sensors in the bioreactor. Wide commercial production of PHA is limited due to its high cost [5], thus improvement in process monitoring with minimal cost is desirable.

Soft sensor is an alternative to laboratory analysis where the terms software and sensor are combined. The principle of soft-sensor is an estimation of a process output variable is obtained using on-line data that are relatively easy to measure or access [6]. The use of soft-sensor facilitates in understanding complex bioprocess dynamics and batch variations [1].

There are two methods for soft sensor development, namely model driven and data driven. Data driven soft sensor are mostly based on First Principle Models that describe physical and chemical background, thus require in depth knowledge of the process [7], [8]. The complex nature of bioprocess requires many assumptions resulting in inaccurate model-based soft sensor.

Data-driven soft-sensor describes the true conditions of the process as it is based on the data collected in the processing plant. Nonlinear data-driven soft sensor models are used in this work to account for process nonlinearity. Also, comparative study on E. coli shows artificial neural network (ANN) which is a nonlinear method outperformed other linear software sensor methods [4].

Artificial neural network has been used on soft-sensor development to estimate process output variables related to product quality. This includes biomass concentration for baker yeast fermentation [9], glucose concentration for secreted protein production [10], biomass, substrate and nosiheptide concentrations in nosiheptide fermentation [11], biomass in nosiheptide fermentation [12], optical density and dry cell weight in recombinant protein production [13] and hydrogen in biological hydrogen production[14]. However, none of these work uses large scale real production data without full consideration of process and measurement uncertainty. Only data from simulation model or mathematical model as well as lab scale production process were used in building ANN soft sensor.

In this work, process data from a pilot scale 2000L bioreactor for PHA production were used in developing soft sensor for optical density estimation using multilayer perceptron neural network (MLP). Preprocessing of process data was done by normalization and interpolation techniques. For comparison with other nonlinear data-driven methods, Radial Basis Function Neural Network (RBFNN) and Support Vector for Regression (SVR) soft sensors were developed.

2.0 Methodology

2.1 Fermentation Process Data

PHA fermentation process was carried out in a 2000L pilot-scale fed-batch bioreactor. The bioreactor was fitted with in-situ optical sensors to measure the process variables such as temperature, pH, agitation speed and dissolved oxygen level. The bioreactor is equipped with a Supervisory Control
and Data Acquisition (SCADA) system. Process data from six different fermentation batches (A to F) were recorded. The fermentation batches vary in length around 21 hours to 45 hours.

The chosen inputs are temperature, agitation speed, pH, and dissolved oxygen (DO) while the output is optical density (OD). These variables have function relationship with optical density, a variable that indicates the amount of biomass in the fermentation broth. In a fermentation process, pH, temperature and dissolved oxygen are important environmental variables that need to be controlled. Low temperature decreases protein’s reaction while high temperature alters the characteristic properties of the protein. For efficient transfer of feedstuff to cell membrane and energy out of the cell, a suitable pH level is required [15]. Output of PHA production is strongly influenced by operating conditions such as temperature, pH level and dissolved oxygen[16], [17]. Dissolved oxygen is a key micro-nutrient for growth, metabolic production and maintenance of the microorganism. Agitation speed adjustments is one of the means for control of DO concentration in a fermentation broth [18].

2.2 Data Pre-processing

Temperature, pH, agitation speed and dissolved oxygen are the on-line process data obtained from Supervisory Control and Data Acquisition (SCADA) system with sampling time of 1 minute. These data are then down sampled to 1 hour/sample so that the sampling frequency difference with offline data is adequately reduced. The process output variable which is optical density values are sampled offline at a lower frequency of 6 hour/sample.

Optical density values are then interpolated using Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) in order to have equal input and output sampling time for use in soft-sensor modelling. PCHIP was chosen rather than linear interpolation since it can interpolate while preserving the shape of data without unnecessary oscillations [19], [20]. Figure 1 shows the comparison between original optical density value, PCHIP interpolation and linear interpolation for Batch A.

The process variables are different in order of magnitude as shown by inputs for Batch C in Figure 2. Normalization was performed to avoid domination of larger magnitude variables on smaller ones and improves the learning process [21], [22]. Thus, all process data are normalized in the range [0.1, 0.9] using equation (1).

\[
X_n = \frac{x_0 - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (2.1)
\]

In equation (1), \(X_n\) is the normalized data, \(x_0\) represents the original data, while \(x_{\text{min}}\) and \(x_{\text{max}}\) are the minimum and maximum value of data, respectively.

![Figure 1: Comparison of interpolation techniques](image-url)
2.3 Multilayer Perceptron Neural Network (MLP) Model Development

A feedforward multilayer perceptron (MLP NN) with Levenberg Marquart (LM) error backpropagation training algorithm was used in modeling optical density. MLP NN was developed in MATLAB R2017b environment. LM is often regarded as the most efficient algorithm in terms of speed of convergence and accuracy in finding the optimal point [23]. It is a three-layer neural network with 4-X-1 structure with X is the number of neurons in hidden layer, determined by trial and error. The output of a hidden neuron in MLP NN is given by Equation 2 [24].

\[ Y_i = f \left( \sum_{j=1}^{N} w_{ij}x_i + b_i \right) \]  \hspace{1cm} (2.2)

In Equation 2, with \( i \) is the number of neurons, \( f \) is the transfer function, \( w \) is the weight, \( N \) is the number of data points, \( x \) is the input and \( b \) is the bias term.

The ANN model has four neurons in the input layer and one neuron in the output layer, equivalent to the number of input and output variables, respectively as shown by model structure in Figure 3. The neural network has one hidden layer with logarithmic sigmoid (logsig) activation function and a linear transfer function (purelin) is used in the output layer. Equation 3 describes the logsig function while purelin function is described by Equation 4.

\[ \text{logsig}(n) = \frac{1}{1 + \exp(-n)} \]  \hspace{1cm} (2.3)

\[ \text{purelin}(n) = n \]  \hspace{1cm} (2.4)

Weight and bias were added to every neuron in hidden layer and output layer. Performance of the ANN model was evaluated using correlation coefficient \( (R^2) \) given by Equation 5. Correlation coefficient value are used to as a measure of difference between the actual and predicted OD values.

\[ R^2 = \frac{\sum(y_{\text{out}} - y_{\text{target}})^2}{\sum(y_{\text{out}} - y_{\text{mean}})^2} \]  \hspace{1cm} (2.5)
Six models were developed using different combination of fermentation batches as training and testing data. The batches are divided as listed in Table 1. Six batches were used for training and one batch for testing.

**Table 1** Division of batch data

| Model Batch | A   | B   | C   | D   | E   | F   |
|-------------|-----|-----|-----|-----|-----|-----|
| Model 1     | Train | Train | Train | Train | Train | Test |
| Model 2     | Test | Train | Train | Train | Train | Train |
| Model 3     | Train | Test | Train | Train | Train | Train |
| Model 4     | Train | Train | Test | Train | Train | Train |
| Model 5     | Train | Train | Train | Test | Train | Train |
| Model 6     | Train | Train | Train | Test | Train | Train |

![Figure 3: MLPNN model structure](image)

### 2.4 Radial Basis Neural Network (RBFNN) Model Development

Radial Basis Function Neural Network (RBFNN) is among the classical feedforward neural network. It has three layers: an input layer, a hidden layer and an output layer [25]. The basic structure of an RBF NN is shown in Figure. The hidden layer uses a nonlinear radial basis activation function and a linear function in the output layer [26]. The output of an RBF NN output neuron is represented by

\[
y_i(x) = \sum_{j=1}^{h} \phi_j(x) \times w_{ij}
\]  

(2.6)

With \( h \) is the number of hidden neuron, \( \phi \) is the radial basis activation function, \( x \) the input and \( w \) is the connecting weight between hidden neuron and output neuron [27]. Gaussian radial basis activation function is used represented by
\[
\phi_j (x) = \exp \left( \frac{\| x - C_j \|^2}{2\sigma_j^2} \right)
\]

(2.7)

\(C_j\) is the center and \(\sigma\) is the spread of radial basis function that represents its width. Spread or width is the value that determines the smoothness of radial basis interpolating function. In order to determine the optimum spread, different values were used in RBF NN training and the network performance was evaluated in terms of its correlation coefficient value as given in Equation (5). RBF NN was developed in MATLAB R2017b environment.

2.5 Support Vector Regression (SVR) Model Development

Support Vector Regression is an adaptation of Support Vector Machine (SVM), a statistical/machine learning theory based classification paradigm [28]. SVR was initially used in pattern recognition problem and later modified to solve regression problems. Through training, nonlinear mapping \(f(x)\) or correlation was obtained between input and output of the learner [29]. The regression function that describes the nonlinear relationship between input and output of the learner is represented by

\[
f(x) = w^T \varphi(x) + b
\]

(2.8)

In Equation (8), coefficients \(w\) and \(b\) need to be adjusted. In this work SVR model was developed in MATLAB R2017b environment and default linear kernel function was used.

3.0 Results And Discussion

3.1 MLP Neural Network Model

For MLPNN, six models were developed using different combination of batch data for training and testing the neural network. Different subset was used for testing in each model to test the generalization capability of the network. The number of hidden neurons was determined by repeated runs or trial and error, varying between 2 to 11 nodes.

Figure 4 (a)-(f) show the correlation coefficient values for different number of hidden nodes for the six models. Higher value of correlation coefficient indicates the model performs better. However, good performance in training subset but poorly in testing subset indicates overfitting. This happens when the number of hidden nodes is too large as can be clearly seen for 11 hidden nodes in all six models. Each model was simulated five times for each number of hidden nodes. The best performance was chosen from these runs. A comparison of best results from each model is shown in Figure 5 and Table 5. The best MLP NN is Model 5 with 10 hidden neurons based on the higher correlation coefficient values for training and testing subsets.
Correlation coefficient

Hidden node

(a)

Model 1

Hidden node

(b)

Model 2

Hidden node

(c)

Model 3

Hidden node

(d)

Model 4

Hidden node
Figure 4: MLPNN correlation coefficient for different hidden node

Figure 5: Comparison of model performance
Table 5 MLPNN model performance

| Model | Hidden node | Train R² | Test R² |
|-------|-------------|----------|---------|
| 1     | 5           | 0.981    | 0.900   |
| 2     | 3           | 0.965    | 0.904   |
| 3     | 9           | 0.991    | 0.933   |
| 4     | 9           | 0.984    | 0.911   |
| 5     | 10          | 0.993    | 0.950   |
| 6     | 10          | 0.986    | 0.948   |

3.2 RBF Neural Network Model

RBF Neural Network models were simulated using different spread values between 30 to 50. Table 6 gives the summary of several trial and error iterations conducted to determine the value of spread that gives the highest correlation coefficient for each model. The best spread value with corresponding correlation coefficient is listed for each model. The best model for RBFNN soft sensor is Model 4 as shown in Figure 6.

| Model No. | Correlation coefficient |
|-----------|-------------------------|
| train     | test                    |
| RBFNN     |                         |

Figure 6: Comparison of model performance for RBFNN

Table 6 RBFNN model performance

| Model | Spread | Train R² | Test R² |
|-------|--------|----------|---------|
| 1     | 30     | 0.971    | 0.869   |
| 2     | 45     | 0.962    | 0.921   |
| 3     | 45     | 0.960    | 0.935   |
| 4     | 45     | 0.961    | 0.951   |
| 5     | 45     | 0.963    | 0.941   |
| 6     | 45     | 0.965    | 0.915   |
3.3 SVR Model

SVR model performances for all models using training and testing subsets are shown in Figure 7 and tabulated in Table 7. Model 4 gives the best performance based on highest correlation coefficient values for training subset and smallest difference with testing subset.

![Comparison of model performance for SVR](image)

**Figure 7** Comparison of model performance for SVR

**Table 7** SVR model performance

| Model | Train R² | Test R² |
|-------|----------|---------|
| 1     | 0.899    | 0.927   |
| 2     | 0.909    | 0.872   |
| 3     | 0.893    | 0.960   |
| 4     | 0.909    | 0.917   |
| 5     | 0.908    | 0.908   |
| 6     | 0.906    | 0.826   |

3.4 Comparison of Soft-sensor Model

Figure 8 shows the comparison of soft-sensor estimation of optical density obtained by MLPNN model, RBFNN model and SVR model for training subset. The results clearly show that optical density estimation by MLPNN soft sensor is very close to the values obtained by offline laboratory analysis. MLPNN soft-sensor gives the most accurate estimation followed by RBFNN while SVR soft-sensor gives the least accurate estimation. Meanwhile, model validation using testing subset shows comparable performance for MLPNN and RBFNN as depicted in Figure 9. All three methods are able to obtain good accuracy with over 0.9 correlation coefficient values.
Figure 8 Comparison optical density estimation using training subset (blue line from lab analysis, red line soft-sensor estimation)

Figure 9 Comparison optical density estimation using testing subset (blue line from lab analysis, red line soft-sensor estimation)

Table 8 Comparison of soft-sensor methods

| Method  | Model | Train $R^2$ | Test $R^2$ |
|---------|-------|-------------|------------|
| MLP NN  | 5     | 0.993       | 0.950      |
| RBFNN   | 4     | 0.961       | 0.951      |
| SVR     | 4     | 0.909       | 0.917      |

4.0 Conclusion

Soft-sensor approach for estimation of optical density values in PHA production achieved a good accuracy in terms of correlation coefficient exceeding 0.9. Three nonlinear data-driven methods were used in building soft sensor model using online process data from a pilot-scale bioreactor. MLPNN soft-sensor performance is superior to both RBFNN and SVR soft sensors. The estimation of optical density is close to those obtained using laboratory analysis. The developed soft sensor could be used to predict process output quality as part of process monitoring and control.
Acknowledgement

This research is fully supported by UPM Putra Grant with grant number 9552500. The authors fully acknowledged Universiti Putra Malaysia for the approved fund which makes this important research viable and effective.

References

[1] P. Biechele, C. Busse, D. Solle, T. Scheper, and K. Reardon, “Sensor systems for bioprocess monitoring,” *Eng. Life Sci.*, vol. 15, no. 5, pp. 469–488, 2015.

[2] K. H. Huong, C. H. Teh, and A. A. Amirul, “Microbial-based synthesis of highly elastomeric biodegradable poly(3-hydroxybutyrate-co-4-hydroxybutyrate) thermoplastic,” *Int. J. Biol. Macromol.*, vol. 101, pp. 983–995, 2017.

[3] M. Von Stosch, “Toward Intensifying Design of Experiments in Upstream Bioprocess Development: An Industrial Escherichia coli Feasibility Study,” *Biotechnology Prog.*, vol. 32, no. 5, pp. 1343–1352, 2016.

[4] K. Kiviharju, K. Salonen, U. Moilanen, and T. Eerikäinen, “Biomass measurement online: The performance of in situ measurements and software sensors,” *J. Ind. Microbiol. Biotechnol.*, vol. 35, no. 7, pp. 657–665, 2008.

[5] J. Mozéjko-Ciesielska and R. Kiewisz, “Bacterial polyhydroxyalkanoates: Still fabulous?,” *Microbiol. Res.*, vol. 192, pp. 271–282, 2016.

[6] R. Luttmann et al., “Soft sensors in bioprocessing: A status report and recommendations,” *Biotechnol. J.*, vol. 7, no. 8, pp. 1040–1048, 2012.

[7] P. Kadlec, B. Gabrys, and S. Strandt, “Data-driven Soft Sensors in the process industry,” *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, 2009.

[8] B. Bidar, J. Sadeghi, F. Shahraki, and M. M. Khalilipour, “Data-driven soft sensor approach for online quality prediction using state dependent parameter models,” *Chemom. Intell. Lab. Syst.*, vol. 162, no. September 2016, pp. 130–141, 2017.

[9] L. Z. Chen, S. K. Nguang, X. M. Li, and X. D. Chen, “Soft sensors for on-line biomass measurements,” *Bioprocess Biosyst. Eng.*, vol. 26, no. 3, pp. 191–195, 2004.

[10] Y. Tian, J. Zhang, and J. Morris, “Optimal control of a fed-batch bioreactor based upon an augmented recurrent neural network model,” *Neurocomputing*, vol. 48, pp. 919–936, 2002.

[11] Q. Yang, H. Gao, W. Zhang, Z. Chi, and Z. Yi, “A new data-driven modeling method for fermentation processes,” *Chemom. Intell. Lab. Syst.*, vol. 152, pp. 88–96, 2016.

[12] Q. Yang and F. Yan, “Soft Sensor of Biomass in Fermentation Process Based on Robust Neural Network,” in *Communications and Information Processing. Communications in Computer and Information Science*, 2012, pp. 273–280.

[13] M. Jenzsch, R. Simutis, G. Eisbrenner, I. Stückrath, and A. Lübbert, “Estimation of biomass concentrations in fermentation processes for recombinant protein production,” *Bioprocess Biosyst. Eng.*, vol. 29, no. 1, pp. 19–27, 2006.

[14] L. M. Rosales-Colunga, R. G. García, and A. De León Rodriguez, “Estimation of hydrogen production in genetically modified E. coli fermentations using an artificial neural network,” *Int. J. Hydrogen Energy*, vol. 35, no. 24, pp. 13186–13192, 2010.

[15] A. Ashoori, B. Moshiri, A. Khaki-Sedigh, and M. R. Bakhtiari, “Optimal control of a nonlinear fed-batch fermentation process using model predictive approach,” *J. Process Control*, vol. 19, no. 7, pp. 1162–1173, 2009.

[16] M. Villano et al., “Effect of pH on the production of bacterial polyhydroxyalkanoates by mixed cultures enriched under periodic feeding,” *Process Biochem.*, vol. 45, no. 5, pp. 714–723, 2010.

[17] W. Shen, A. Seak, M. Chua, H. Koon, T. Nittami, and G. Cheng, “Strategy for the biotransformation of fermented palm oil mill effluent into biodegradable polyhydroxyalkanoates by activated sludge,” *Chem. Eng. J.*, vol. 269, pp. 288–297, 2015.
[18] L. Bodizs, M. Titica, N. Faria, B. Srinivasan, D. Dochain, and D. Bonvin, “Oxygen control for an industrial pilot-scale fed-batch filamentous fungal fermentation,” J. Process Control, vol. 17, no. 7, pp. 595–606, 2007.
[19] M. Abbas, A. A. Majid, M. N. H. Awang, and J. M. Ali, “Positivity-preserving C2 rational cubic spline interpolation,” ScienceAsia, vol. 39, no. 2, pp. 208–213, 2013.
[20] S. Butt and K. W. Brodlie, “Preserving Positivity Using Piecewise,” Comput. Graph., vol. 17, no. 1, pp. 55–64, 1993.
[21] M. Shafaei and O. Kisi, “Predicting river daily flow using wavelet-artificial neural networks based on regression analyses in comparison with support vector machine models,” Neural Comput. Appl., vol. Article in, pp. 1–14, 2016.
[22] Z. Zainuddin, W. R. Wan Daud, O. Pauline, and A. Shafie, “Wavelet neural networks applied to pulping of oil palm fronds,” Bioresour. Technol., vol. 102, no. 23, pp. 10978–10986, 2011.
[23] M. G. Moghaddam, F. B. H. Ahmad, M. Basri, and M. B. A. Rahman, “Artificial neural network modeling studies to predict the yield of enzymatic synthesis of betulinic acid ester,” Electron. J. Biotechnol., vol. 13, no. 3, pp. 1–12, 2010.
[24] E. Heidari, M. A. Sobati, and S. Movahedirad, “Accurate prediction of nanofluid viscosity using a multilayer perceptron artificial neural network (MLP-ANN),” Chemom. Intell. Lab. Syst., vol. 155, pp. 73–85, 2016.
[25] X. Ru-Ting and H. Xing-Yuan, “Predictive calculation of carbon dioxide solubility in polymers,” RSC Adv., vol. 5, no. 94, pp. 76979–76986, 2015.
[26] M. D. Buhmann, Radial Basis Functions: Theory and Implementations. Cambridge University Press, 2003.
[27] V. Agarwal, “Radial basis function neural network-based face recognition using firefly algorithm,” Neural Comput. Appl., vol. 30, no. 8, pp. 2643–2660, 2018.
[28] V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer Science + Business Media, 2000.
[29] A. Kavousi-Fard, H. Samet, and F. Marzbani, “A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting,” Expert Syst. Appl., vol. 41, no. 13, pp. 6047–6056, 2014.