

## Realization of Deep Learning Based Embedded Soft Sensor for Bioprocess Application

V. V. S. Vijaya Krishna<sup>1,\*</sup>, N. Pappa<sup>1</sup> and S. P. Joy Vasantharani<sup>2</sup>

<sup>1</sup>Department of Instrumentation Engineering, MIT, Anna University, Chennai, India

<sup>2</sup>Department of Electronics Engineering, MIT, Anna University, Chennai, India

\*Corresponding Author: V. V. S. Vijaya Krishna. Email: v.vijayakrishna@gmail.com

Received: 30 July 2021; Accepted: 02 September 2021

**Abstract:** Industries use soft sensors for estimating output parameters that are difficult to measure on-line. These parameters can be determined by laboratory analysis which is an offline task. Now a days designing Soft sensors for complex nonlinear systems using deep learning training techniques has become popular, because of accuracy and robustness. There is a need to find pertinent hardware for realizing soft sensors to make it portable and can be used in the place of general purpose PC. This paper aims to propose a new strategy for realizing a soft sensor using deep neural networks (DNN) on appropriate hardware which can be referred as embedded soft sensor (ESS). The work focuses on developing an ESS for estimating lactose concentration in a simulated and experimental bioreactor using DNN and realizing it on the Zynq based System on Chip (SoC). Deep neural network is developed for the process with certain number of hidden layers. The model parameters of the process is represented at input layer and lactose concentration is considered at output layer. The performance of the ESS has been observed with the number of hidden layers and different activation functions. Then the optimized neural network is chosen for realizing on hardware. Comparison is made among the values obtained from hardware realization, software simulation and laboratory analysis. Output analysis shows that the values obtained through hardware realization are closer to the values obtained through laboratory analysis. From the results it can be concluded that Deep learning provides a better way, alternative to traditional techniques for realizing ESS on hardware. From the proposed work, it can be shown that if any sensor is unavailable for measuring any parameter then this ESS can be used to measure the values. Since this ESS is realized on reconfigurable hardware like SoC, it can be portable and flexible to measure values.

**Keywords:** Soft sensor; embedded soft sensor; deep neural networks; bioreactor; zynq; system on chip; bioreactor; reconfigurable hardware

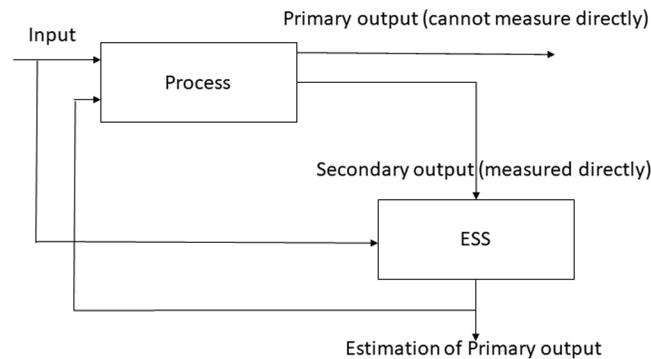


This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

Industries with control and processing units are largely equipped with bulk instruments and more number of sensors. Sensors are used to send data to different units for processing. There may be situations where a hardware sensor may not be available or the sensor may be expensive. In such situations, the data from available sensors are used to build models for processing units based on estimation algorithms. These models can be referred as Soft Sensors (SS) which are software programs that give same information as compared to hardware counterparts. There are several number of process industries like chemical plants, Refining industries, Environmental based organizations, etc., which are using Soft Sensors. For example, Soft Sensors is used for estimating the kerosene freezing point [1] in an atmospheric distillation unit in which a Neural Network-Principal Component analysis (NN-PCA) is used to realize on Field Programmable Gate Array (FPGA), estimating sulphur in [2] Sulphur recovery unit (SRU) for measuring  $H_2S$  from gas streams, etc. Soft Sensors uses estimation algorithms which [3] are iterative algorithms where final values of the variables are calculated based on assumed initial values and specifications of the system. This leads to automation of the process where manual measurement is replaced by soft sensor [4,5]. There is a need to identify appropriate hardware for soft sensors to make it portable, so that it can replace a conventional PC for an automatic process system. Hence it is aimed to propose a new strategy for the hardware realization of a soft sensor on a reconfigurable device like System on Chip (SoC).

A general abstract model of the process can be represented by two types of outputs, like primary and secondary. Secondary outputs can be measured directly with available hardware sensors whereas primary outputs cannot be measured directly [6]. This may be due to unavailability of hardware sensors, sensors may be costly, available sensors may be with limited measurements, etc. [7,8]. Fig. 1 represents the system where a proposed Embedded Soft Sensor (ESS) is used for calculating the primary outputs from secondary outputs.



**Figure 1:** Block diagram representing the function of ESS

In the block diagram an embedded soft sensor has one input which is a secondary output from process. The secondary outputs are measured using available hardware sensors. An embedded soft sensor which is proposed here, is to measure the Primary output which cannot be measured using hardware sensors. It is an iterative process and initial value is assumed. The initial assumption depends on process parameters and it leads to correct the estimated value during the course of certain iterations [9]. The estimated output can be verified with laboratory analysis result. In this paper a bioprocess application is considered for examining the performance of ESS which is realized on a reconfigurable hardware device like SoC by a hardware. For designing the SS there are various techniques available in literature. Some of them are support vector systems, neural networks, fuzzy systems, principal component analysis, etc. Though these

techniques are considered as mainstream algorithms, they cannot guarantee an optimized design for highly nonlinear design, systems with high complexity and handling large data sets, data drift, missing data, etc. Since DNN provides a way for handling some of the above problems, proposed work is based on DNN to achieve less error and high accuracy for nonlinear systems.

In recent days, DNN algorithms are widely used in applications like voice recognition, machine vision system, intelligent retrieval, virtual assistance, bioprocessing, etc. DNN techniques are part of machine learning methods which comprises of two or more hidden layers. DNN with single layer is called shallow neural network (SNN). In SNN, features can be learned [10] independently with arbitrary functions and considerable number of neurons in hidden layer. As number of hidden layers are increased, large data set can be processed and better accuracy of the system [11] can be achieved.

Developing DNN for bioprocess applications are not on par with other fields. This may be due to highly nonlinear functions, dependency on past data, and time for yield. Generally these processes are slow and run in batches. This data has to be processed to estimate the unknown parameter. This entire data acquisition process is carried off-line. Some data may be unlabelled and this is also to be considered for estimating unknown parameter [11]. Major difficulty in this bioprocess is gradual decline of accuracy. This may be due to sudden change in process variables of the system, change in chemical or biological reaction, performance degradation of catalyst, drift in sensor data, etc. These problems can be handled by DNN architectures because, the weight of each neuron can be updated after checking the error at the output. Back propagation algorithm includes automatic weight update based on the error and can be continued for certain number of iterations until error is minimized. Therefore DNN architectures are suitable to design a nonlinear process which is difficult to acquire the data, design of systems with unlabelled data, design of batch processing systems, design of systems where time varying parameters are critical, design of systems where online measurement is difficult [12]. Even though the design of SS is achieved, identifying and realizing on appropriate hardware is also important for making the SS portable. Realizing on programmable hardware makes SS more easy and robust.

In recent year's very large scale integration (VLSI) technology is widely used in bioprocess applications. Nowadays SoCs are gaining more importance as they are considered as an alternative to traditional FPGAs [13]. They are classified as heterogeneous devices as they have both the processor and FPGA integrated on a single chip. Hardware/Software (HW/SW) mechanism implies to design models at various levels of abstraction ranging from gate level to algorithmic level. Software programs based on high level languages also can be used to design the process models [14]. Even though DNN is explored in various applications, bioprocess is still behind because of its challenges in representing a complex nonlinear system and highly varying properties [15]. There is a high need for hardware realization as it makes more portable and robust.

Hence, in this paper the work is focussed on realizing a DNN based ESS for measuring lactose concentration on a SoC hardware. Section 2, describes the bioreactor, its specifications, mathematical background. In Section 3, SS design using appropriate DNN algorithm is presented Section 4 presents proposed system architecture and implementation of ESS on SoC hardware. Section 5 describes hardware realization results and comparison. Conclusion and future work is presented in Section 6.

## 2 Bioreactor

The application considered in this paper is bioprocess. In this process an enzyme,  $\beta$ -galactosidase (lactose) or substrate is produced in a bioreactor. This enzyme is used widely in food industry. Bioreactor is a certain type of vessel where cultivation of cell is performed under futile and controlled conditions. The process is a batch process which means the inputs are fed in batches and outputs are sampled at certain intervals of time. The main characteristic of this batch process is the substrate (lactose)

concentration continuously decreases and concentration of biomass and product will tend to increase. Bioreactor works in controlled environment which depends on controlled variables for producing different bio products which are used in process industry [16]. To observe controlled environment, some vital variables are to be continuously checked and controlled. Hardware sensors are used to acquire the values of these variables. Variable values which are difficult to measure using hardware sensors are estimated through soft sensors [17]. In the present paper, an approach is proposed to design and realize an embedded soft sensor for estimating lactose concentration in a bioreactor using DNN algorithms. For training and testing the neural network, the data is obtained from the experimental values from bioreactor [18] which is based on the following model equations.

$$\frac{dX}{dt} = r_{\text{ferm}} + r_{\text{oxid}}^s + r_{\text{oxid}}^{\text{Et}} - r_d \quad (1)$$

$$\frac{dS}{dt} = - \frac{1}{Y_{X/S}^{\text{ferm}}} \cdot r_{\text{ferm}} - \frac{1}{Y_{X/S}^{\text{oxid}}} \cdot r_{\text{oxid}}^s \quad (2)$$

$$\frac{dEt}{dt} = \frac{\varnothing_{X/S}^{\text{ferm}}}{Y_{X/S}^{\text{ferm}}} \cdot r_{\text{ferm}} - \frac{1}{Y_{X/E}^{\text{oxid}}} \cdot r_{\text{oxid}}^{\text{Et}} \quad (3)$$

$$\frac{dC_L}{dt} = K_{La} \cdot (C_{Le} - C_L) - \left[ \frac{\varnothing_{X/S}^{\text{oxid}}}{Y_{X/S}^{\text{oxid}}} \cdot r_{\text{oxid}}^s + \frac{\varnothing_{X/E}^{\text{oxid}}}{Y_{X/E}^{\text{oxid}}} \cdot r_{\text{oxid}}^{\text{Et}} \right] \quad (4)$$

$$\frac{dC_G}{dt} = - \frac{F_{ar} \cdot C_G}{V_G} + \frac{F_{ar} \cdot C_{Go}}{V_G} - K_{La} (C_{Le} - C_L) \cdot \frac{V_L}{V_G} \quad (5)$$

where

X is the concentration of living cells [Biomass Concentration] [ $\text{g dm}^{-3}$ ]

S is the concentration of Lactose [ $\text{g dm}^{-3}$ ]

Et is the concentration of ethanol [ $\text{g dm}^{-3}$ ]

$C_L$  is the concentration of dissolved oxygen in the liquid phase [ $\text{g dm}^{-3}$ ]

$C_G$  is the concentration of oxygen in the gas phase [ $\text{g dm}^{-3}$ ]

$r_{\text{ferm}}, r_{\text{oxid}}^s, r_{\text{oxid}}^{\text{Et}}, r_d$  are the lactose-Limited Specific growth rates [ $\text{g dm}^{-3} \text{h}^{-1}$ ]

$Y_{X/S}^{\text{ferm}}, Y_{X/S}^{\text{oxid}}, Y_{X/E}^{\text{oxid}}$  are the yield coefficients [dimensionless]

$\varnothing_{X/S}^{\text{ferm}}, \varnothing_{X/S}^{\text{oxid}}, \varnothing_{X/E}^{\text{oxid}}$  are the stoichiometric values related to bioprocess [dimensionless]

$K_{La}$  is gas-liquid mass transfer coefficient of the oxygen [ $\text{h}^{-1}$ ]

$C_{Le}$  is the liquid phase oxygen concentration in equilibrium with the gas [ $\text{g dm}^{-3}$ ]  $F_{ar}$  is their flow rate injected in the bioreactor [ $\text{dm}^3 \text{h}^{-1}$ ]

$V_G$  is the internal volume of gas in the bioreactor [ $\text{dm}^3$ ]

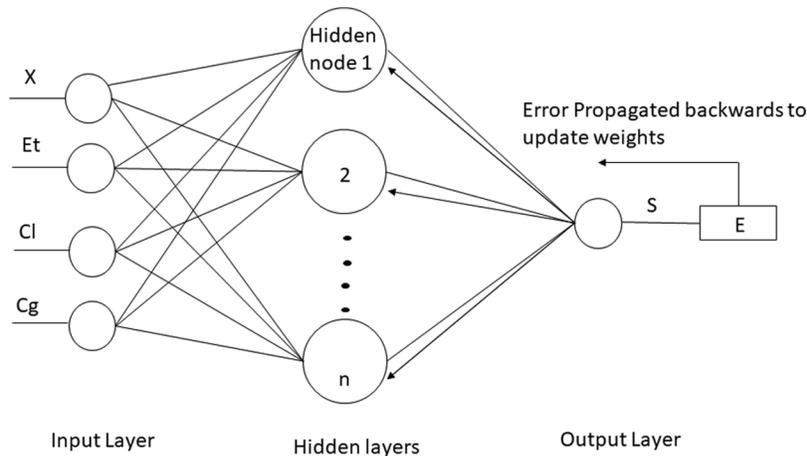
$V_L$  is the volume of liquid of the bioreactor [ $\text{dm}^3$ ]

$C_{Go}$  is the oxygen concentration in the air feeding flow [ $\text{g dm}^{-3}$ ],

t is time [h]

### 3 Soft Sensor Design Using DNN

In this section, different deep forward neural networks are developed based on activation functions and the number of hidden layers. Back propagation algorithm is applied on these networks to obtain minimum error of the output parameter. Fig. 2 represents a general neural network with backpropagation algorithm to estimate the lactose concentration in a bioreactor. There are five inputs at the input layer and one output at the output layer. Different number of hidden layers and activation functions can be used.



**Figure 2:** Neural Network representation for estimating Lactose concentration

At each neuron or node of the network, the weights are calculated and this process is performed up to neurons of output layer. The weight of the  $k^{\text{th}}$  neuron of  $i^{\text{th}}$  hidden layer is indicated as  $w_{kj}^i$ . The value of  $k$  range from 1 to  $n$ . The input to  $k^{\text{th}}$  neuron of  $i^{\text{th}}$  hidden layer is calculated by the following equation.

$$\sum_j w_{kj}^i \cdot a_j + b_k^i = \text{net}_k^i \quad (6)$$

The estimated output is calculated at the output neuron and is compared with the true value. The error between estimated and true value is propagated back to the previous layer and the weights are recalculated. This iteration is continued until the calculated error is minimized. The process of updating weights depending on the previous error is called back propagation algorithm.

Weight initialization is to be carried out for preventing activation outputs from vanishing during the forward pass from input layer to output layer of the neural network. If vanishing problem occurs, then loss gradient will become small and it is difficult to flow backwards for calculating the neuron input. Hence network will take more iterations to converge to optimized output [19]. If weights of neurons of input layer are initialized to 0 then, the loss function derivative will be same for every weight and thus all weights will be same for subsequent iterations. Due to this, all hidden layers will be symmetric and same will be continued for other iterations. Hence appropriate weight initialization [20] should be done. The Activation functions are used for normalizing all inputs that are given to different layers of neural network. Normalization makes weight initialization and update less computationally simple and fast during training the network.

The error calculated at the neuron of output layer is given by the following equation where  $n$  represents number of hidden layers used in neural network.

$$E = \frac{1}{2} \sum_k (c - a_k^n)^2 \quad (7)$$

The Weights are updated by:

$$w^{i+1} = w^i + [j^i]^{-1} j^i e \quad (8)$$

The performance of DNN depends on the type of data sets and number of data sets used on network for training. If network is trained properly the accuracy increases. In general if a network tries to remember training data, generalization is lost and over fitting problem occurs. Early stopping [21,22] is used to avoid this problem where the validity error is also calculated in addition to monitoring error. As number of iteration increases, the validation error decreases. With the increase in iterations, the error decreases and tends to increase after some point. At this point the accuracy of network may decline. This point where error reaches minimum is called early stop point. This point of iteration gives the optimized output for the estimated parameter. The number of hidden layers also play an important role during error calculation. Under fitting or over fitting problems may occur if hidden layers are not chosen properly. For updating weights the learning rate is used and it can affect the error and number of iterations.

In brief the soft sensor design can be achieved from the following steps

- A) Select process parameters from the system model.
- B) Segregate and identify training and testing data.
- C) Weight initialization of input neurons.
- D) Evaluate the neuron outputs using forward pass.
- E) Calculation of output error and back propagate it for weight update.
- F) Testing soft sensor with test data.

In this paper, different algorithms based on activation function and number of hidden layers and different weight initialization techniques are used for estimating lactose concentration in a bioreactor. This is represented in Tab. 1. Here 3, 4 and 5 layers are chosen and activation functions like hyperbolic tangent sigmoid (tansig) and rectified liner unit (ReLU) are used. He and Xavier weight initialization techniques are used. Combination of these layers and activation functions and weight initializations will affect the accuracy of estimation. There should be a limit in the number of hidden layers as more layers may result in over fitting problem.

**Table 1:** Summary of DNN algorithms for Estimating Lactose Concentration

Parameter	Alg1	Alg2	Alg3	Alg4
Weight Initialization	Xavier	He	He	He
No of Hidden Layers	3	4	4	5
Activation function (output Layer)	Tansig	Tansig	ReLU	ReLU

The equation for the hyperbolic tangent sigmoid activation function is given as:

$$f(x) = \frac{2}{1 + 2e^{-2x}} - 1 \quad (9)$$

The equation for the Rectified Linear unit is given as:

$$f(x) = \max(0, x) \tag{10}$$

In the neural network, a fivefold testing technique is used. Five equal groups are created from the dataset. Among these groups one group is used for testing and other four groups are reserved for network training. Error at output is represented as root mean square error and is calculated using the following equation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_p} (y - y_i)^2}{N_p}} \tag{11}$$

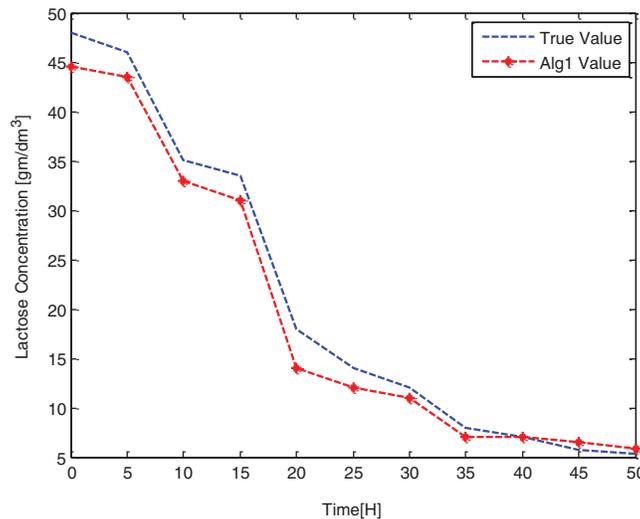
where  $y, y_i$  are the desired output value and the estimate calculated from the network.  $N_p$  represents output layer neurons. An average value is calculated to check the network efficiency.

The codes for all the algorithms are compiled and verified on Python IDE. [Tab. 2](#) refers to the error generated at the output by using different algorithms. From the table it is evident that among the proposed algorithms, Alg4 results in less error when compared to that of algorithms.

**Table 2:** Performance comparison of RMSE value of different algorithms for estimating Lactose concentration

Parameter	Alg1	Alg2	Alg3	Alg4
RMSE value	0.081	0.072	0.0689	0.0465

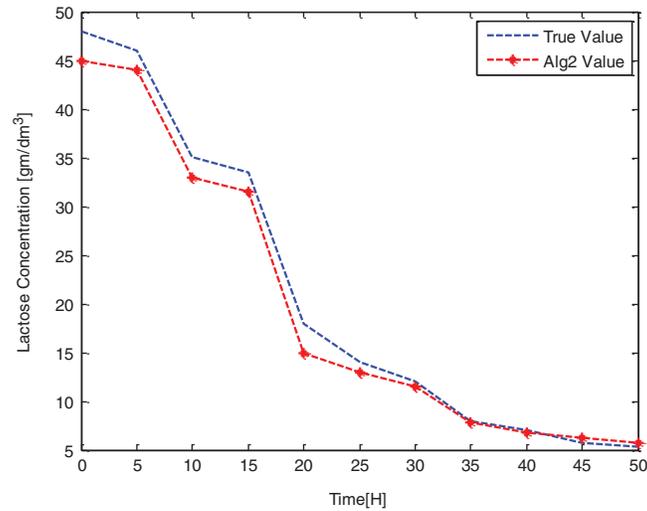
[Figs. 3 to 6](#) represents comparison curves of true value and estimated values of lactose concentration for the proposed 4 algorithms.



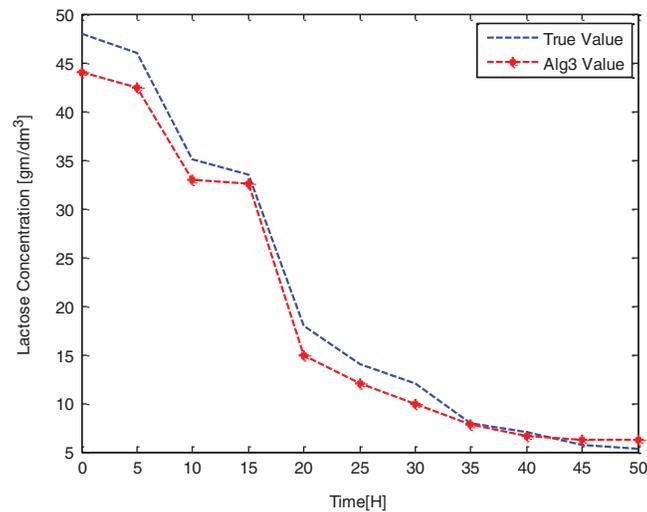
**Figure 3:** Performance curves of lactose concentration from measured values and estimated values from alg1

From the above figures, it is evident that estimated values are closer to measured values for alg4, which is characterized by 4 hidden layers with ReLu activation function and Xavier initialization. It is also clear that the estimated values becomes closer to measured values only after certain iterations. Hence alg4 is

considered for realizing on hardware. The next section of this paper deals with hardware realization of alg4 on Zynq SoC.



**Figure 4:** Performance curves of lactose concentration from measured values and estimated values from alg2



**Figure 5:** Performance curves of lactose concentration from measured values and estimated values from alg3

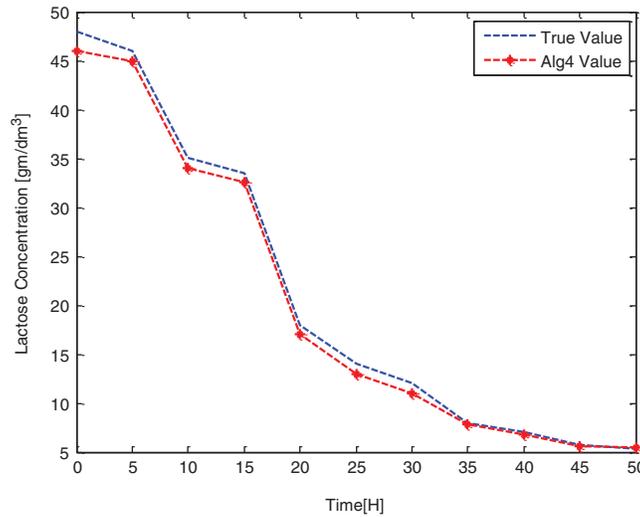
## 4 Hardware Design

The soft sensor designed in previous section is to be realized on a hardware. Here a Zynq SoC is chosen. It has features of both hardcore processor and field programmable gate array.

### 4.1 Zynq SoC

Zynq device is selected for realizing the ESS for bioreactor, because of its resilience and appropriateness towards Hardware/Software approach. The Zynq device integrates a conventional FPGA [23] and a dual core ARM cortex processor on to same chip. The architecture of FPGA is based on Artix family of Xilinx device,

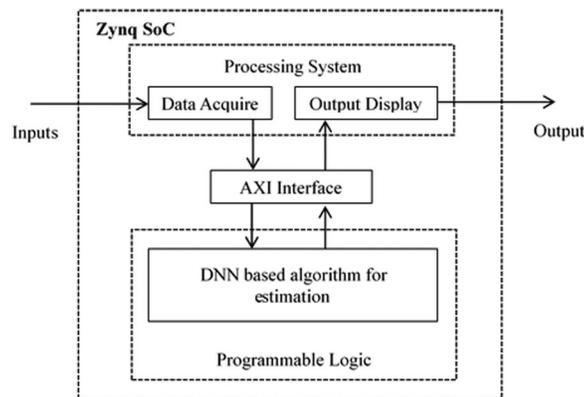
which constitutes as Programmable Logic and ARM processor is considered as Processing System (PS). Integration of both PL and PS allows designing the system at various level of abstraction by optimizing the parameters like speed, power, area etc. when compared to FPGA and Processor realization.



**Figure 6:** Performance curves of lactose concentration from measured values and estimated values from alg4

#### 4.2 System Design Architecture

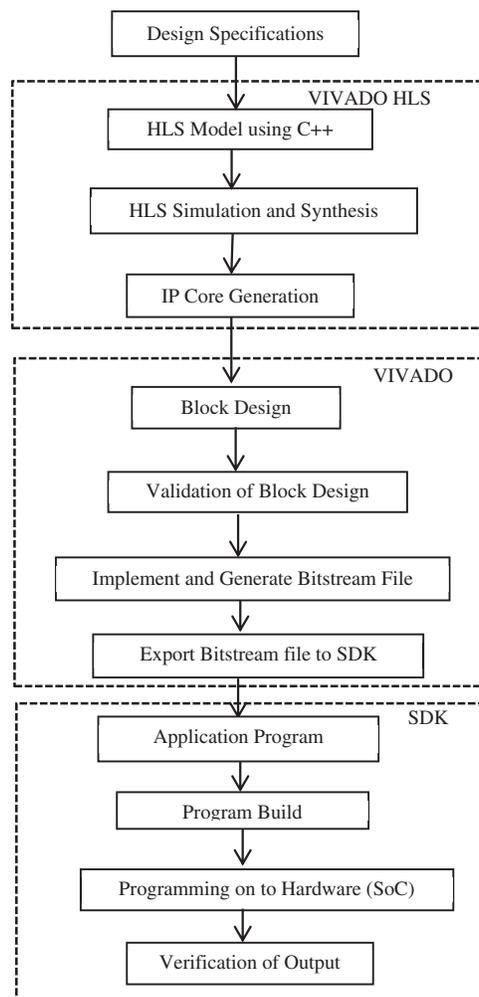
The Simplified system architecture of the embedded soft sensor for bioreactor on Zynq platform using Hardware/Software co-design is shown in Fig. 7. The actual logic for estimating lactose concentration using DNN algorithm is realized as an Intellectual Property (IP) Core using C++ code in PL part. The IP core generated in PL part is used to construct the overall design of soft sensor using IP integrator. The Data acquisition of inputs to soft sensor and displaying the result will be carried using C code which will be processed in PS part. The interconnection between PS and PL can be carried in different ways and in this paper Advanced Extensible Interface (AXI) is used.



**Figure 7:** Block diagram of Zynq SoC Hardware for proposed ESS

### 4.3 Design Flow

The hardware implementation of embedded soft sensor of the bioreactor using DNN has number of steps. The description of the design flow is given in Fig. 8. In brief, an IP core is developed by writing C++ Code in VIVADO HLS, which is validated with corresponding true values. Next a block level design is created with required IPs and that design is validated and a bit stream file is generated. Using Software development Kit (SDK) an application program is developed with respect to soft sensor. Finally, the block level design is realized on hardware and the corresponding application program is executed using the programmed hardware. The design and implementation of soft sensor are carried out using 2016.3 versions of HLS, Vivado and SDK tools [24,25]. The hardware used is Zybo prototyping board which has Zynq device with number as XC7Z010-1CLG400C.



**Figure 8:** Hardware design flow for realizing ESS

#### 4.3.1 IP Design Using Vivado HLS

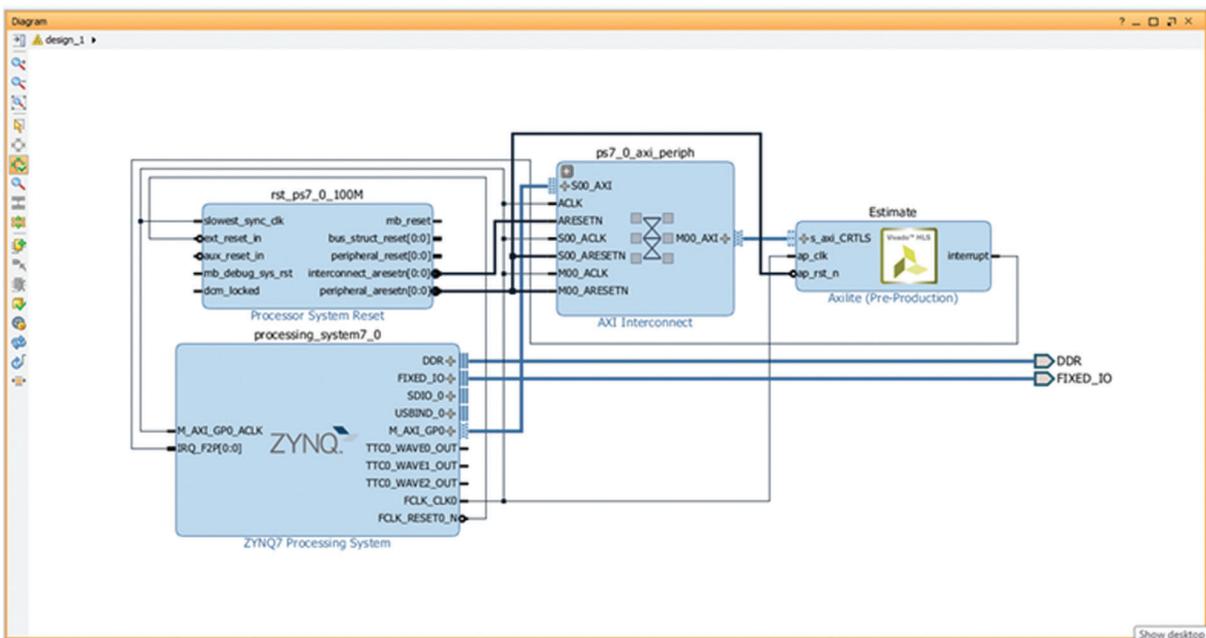
The specifications described initially are used to develop a C++ source code using VIVADO HLS tool. This code is the estimation algorithm for estimating Lactose concentration in a bioreactor.

Directives are used for optimizing the design. The directives used are unrolling loops, and pipelining technique. Loops are applied with “Unroll” directive where the next state of the estimation algorithm is

calculated. This directive makes loop to execute in parallel by having a particular hardware resource for each loop. For parallel execution of top level block “Estimate”, the “Pipeline” directive is used. This allows execution of various sub blocks in top function to run in order. For connecting different IP cores and Zynq PS, “AXI Lite” directive is used. It follows separate communication protocol for transmitting data between different IP blocks. The developed code is compiled, simulated and synthesized using the VIVADO simulator. The synthesized file is used for generating IP Core which can be used in the next step.

#### 4.3.2 Block Design Using Vivado

The block level design of embedded soft sensor using DNN is shown in Fig. 9. While creating the block level design using IP Integrator, the Zynq based PS IP and the HLS based IP (Estimate) are added from the IP list. The remaining IP's are automatically created in the design during block and connection step. These IP's are Processor System Reset and AXI Interconnect. In total, 4 IPs are used. Zynq IP is the “ZYNQ7 Processing System (PS)”, which corresponds to the dual core ARM processor. User defined IP is “Estimate” which is designed simulated and exported from VIVADO HLS. PS and PL are connected using “AXI Interconnect” IP. For implementing global resets across the block level design, “Processor System reset” IP is used.



**Figure 9:** Block level design of ESS on Zynq SoC

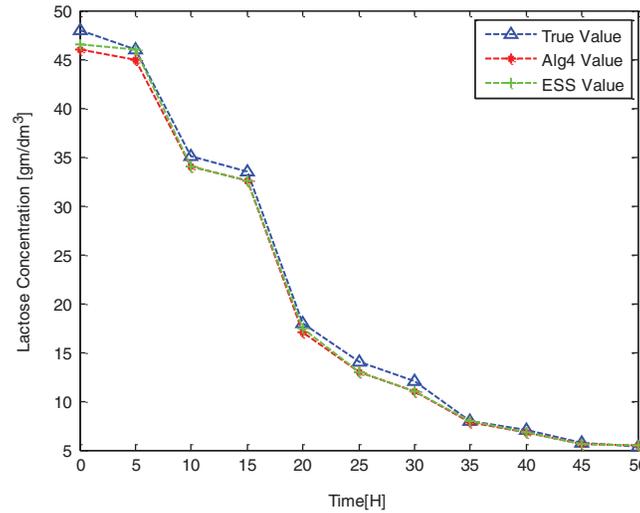
The Block design is validated for errors in block level and after validation the design file is wrapped and synthesized. The validated block is implemented and the bit file is generated which is suitable for programming in to SoC. The generated Bitfile is exported to SDK for programming and verification part.

#### 4.3.3 Application Model Using SDK

Application programs that are to be executed on designed hardware or initializing the settings of hardware can be created using SDK. These are the software programs written in either C or C++. The developed C Code is verified for errors and an ELF file is generated which is used for programming the SoC.

## 5 Results and Analysis

After programming the alg4 on Zynq Soc hardware, the values are recorded and compared with that of the true value and alg4 python compiled values. The comparison curves are represented in Fig. 10. From the figure it is evident that the hardware values of Alg4 are closer to true values.



**Figure 10:** Performance estimation curves of DNN based ESS

Hardware usage and performance of the implemented ESS are shown in Tab. 3. This table gives the information of hardware resources utilized by Zynq SoC under different optimization directives running at maximum frequency of 114 MHz. From the table it is clear that in terms of latency, pipelining and array partitioning directives gives optimized value compared to other.

**Table 3:** Hardware details of ESS

Hardware resources	Unroll directive	Pipeline and array directive
BRAM_18K	12	13
DSP 48E	78	76
FF	34058	33569
LUT	14274	15423
Latency	12232	11456

On-chip power consumption can be represented device static and the dynamic power consumption. Transistor leakages are considered as static power dissipation, which is dependent on device. The power consumed by clocks, Signals, Logic, BRAM, DSP, PS, etc. considered as dynamic power consumption. Tab. 4 represents the power consumption information of Zynq Device.

**Table 4:** Power results of ESS

	Power (W)
Device static	0.134
Clocks	0.015
Signals	0.005
Logic	0.004
BRAM	0.004
DSP	0.001
PS7	1.561
Total	1.724

## 6 Conclusion and Future Work

In this paper, Deep Neural Network based Embedded Soft Sensor for inferring Lactose concentration in a bioreactor is presented. The summary of the work is to devise a soft sensor on a reconfigurable hardware which can be considered as embedded soft sensor. The variables which cannot be measured directly can be measured with this soft sensor. This soft sensor is a set of instructions which are used to estimate the unmeasured variables. The work is further carried to implement these instructions on a reconfigurable hardware like SoC instead of PC. This results in portability and flexibility, since these devices are single chip with flexible features and parallel execution. The Soft Sensor is realized on a SoC chip that can be used in the field or control room of an industry. Functional simulation of embedded soft sensor is compared with the values that are realized on Zynq SoC device. Results indicate that the implemented embedded soft sensor performance is on par with actual laboratory values and simulation values. In future, the integration of multiple sensors which is also called the sensor fusion is to be carried to achieve enhanced reliability, extended parameter coverage and improved resolutions which are all desirable in any system. So the advantage of using embedded technology for realizing Soft sensor allows industries to enhance their performance by reducing functional costs and improving product quality.

**Acknowledgement:** The authors would like to thank Anna University and also like to thank Anonymous reviewers for their so-called insights.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] R. Caponetto, G. Dongola, A. Gallo and M. Xibilia, "Fpga based soft sensor for the estimation of the kerosene freezing point," in *Proc. IEEE Int. Symposium on Industrial Embedded System, SIES'09*, Lausanne, Switzerland, pp. 228–236, 2009.
- [2] L. Furtuna, A. Rizo, M. Sinetra and M. Xibilia, "Soft analysers for a sulphur recovery unit," *Control Engineering Practice*, vol. 11, pp. 1491–1500, 2003.
- [3] J. Manninen and R. Ritala. "Use of modelling and simulation in the pulp and paper industry," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 15, no. 5, pp. 409–423, 2009.
- [4] H. Kaneko and K. Funatsu, "Preparation of comprehensive data from huge data sets for predictive soft sensors," *Chemometrics and Intelligent Laboratory Systems*, vol. 153, pp. 75–81, 2016.

- [5] B. Lin, B. Recke, J. Knudsen and S. Jorgensen, "A systematic approach for soft sensor development," *Computers Chemical Engineering*, vol. 31, pp. 419–425, 2007.
- [6] D. Riordan and J. Walfh, "A cost-effective and accurate electrical impedance measurement circuit design for sensors," *International Journal on Smart Sensing & Intelligent Systems*, vol. 9, no. 2, pp. 28–45, 2016.
- [7] N. Bachtadze, V. Kulba and G. Maximov, "Identification based approach to soft sensors design," *International Federation of Automatic Control Proceedings*, vol. 40, no. 3, pp. 87–92, 2007.
- [8] Y. Shardt and X. Yang, "Development of soft sensors for the case where the time delay is random," *International Federation of Automatic Control Papers Online*, vol. 49, no. 7, pp. 1193–1198, 2016.
- [9] P. Kadlec, B. Gabrys and S. Strandt, "Data driven soft sensors in the process industry," *Computers Chemical Engineering*, vol. 33, no. 4, pp. 795–814, 2009.
- [10] V. Gopakumar, S. Tiwari and I. Rahman, "A deep learning based data driven soft sensor for bioprocesses," *Biochemical Engineering Journal*, vol. 136, pp. 28–39, 2018.
- [11] E. Aljuwaiser, R. Badr and M. Hassan, "A New approach of state estimation of linear discrete systems," *Control Engineering and Applied Informatics*, vol. 20, pp. 3–13, 2018.
- [12] A. Taghiyev, A. Altun, N. Allahverdi and S. Caglar, "A machine learning framework to identify the causes of HBA1C in patients with type 2 diabetes mellitus," *Control Engineering and Applied Informatics*, vol. 21, pp. 34–42, 2019.
- [13] A. Ali, H. Djelouat, A. Amira, F. Bensaali, M. Benammar *et al.*, "Electronic nose system on the zynq SOC platform," *Microprocessors and Microsystems*, vol. 3, pp. 145–156, 2017.
- [14] M. Ning, W. Shaojun and P. Yu, "Implementation of LS-sVM with HLS on zynq," in *Proc. Int. Level Conf. on Control Automation Systems*, Seoul, Korea, pp. 346–349, 2014.
- [15] N. Pletschen and K. Diepold, "Non-linear state estimation for suspension control applications: A takagi-sugeno kalman filtering approach," *Control Engineering Practice*, vol. 61, pp. 292–306, 2017.
- [16] J. Singh, N. Kaushik and S. Biswas, "Bioreactors technology design analysis," *The Scitech Journal*, vol. 1, no. 6, pp. 28–36, 2014.
- [17] A. DeAsis and R. MecielFilho, "Soft sensors development for on-line bioreactor state estimation," *Computers and Chemical Engineering*, vol. 24, pp. 1099–1103, 2000.
- [18] L. Longhi, D. Luvizetto, L. Ferreira, R. Rech, M. Ayub *et al.*, "A growth kinetic model of *kluveromyces marxianus* cultures on cheese whey as substrate," *Journal of Industrial Microbiology and Biotechnology*, vol. 31, no. 1, pp. 35–40, 2004.
- [19] S. Webb, "Deep learning for biology," *Nature*, vol. 554, no. 7690, pp. 554–558, 2018.
- [20] C. Long and H. Nguyen-Xuan, "Deep learning for computational structural optimization," *International Society for Automation Transactions*, vol. 103, pp. 177–191, 2020.
- [21] V. G. Maltarollo, K. M. Honrio and A. B. F. da Silva, "Applications of artificial neural networks in chemical problems," *Artificial Neural Networks Architectures and Applications*, vol. 25, no. 3, pp. 203–224, 2013.
- [22] M. Konishi, "Bioethanol production estimation from volatile compositions in hydrolysates of lignocellulosic biomass by deep learning," *Computers Chemical Engineering*, vol. 129, no. 6, pp. 723–729, 2020.
- [23] L. Crockett, R. Elliot, M. Enderwitz and R. Stewart, "Principles of Hardware-Software Co-design," in *the Zynq Book*, Strathclyde Academic Media, Glasgow, UK, 2014.
- [24] Xilinx, "Embedded Processor Design (UG898)," in *Vivado Design Guide*, Xilinx, 2014.
- [25] Xilinx, "High-Level Synthesis (UG902)," in *Vivado Design Guide*, Xilinx, 2015.