Automatic Implementation of Neural Networks Through Reaction Networks—Part II: Error Analysis

Yuzhen Fan, Xiaoyu Zhang, Chuanhou Gao, *Senior Member, IEEE*, and Denis Dochain, *Fellow, IEEE*

***Abstract*—Neural network related machine learning algorithms, inspired by biological neuron interaction mechanisms, are advancing rapidly in the field of computing. This development may be leveraged in reverse to advance synthetic biology progress. A challenging exploration is to implement neural network functionalities through biochemical reaction networks (BCRNs), a language that is inherently compatible with in vivo, with difficulties specifically in constructing an appropriate BCRN that respects computation and information processing steps involved in neural networks. By addressing these difficulties, this paired article manages to develop a biochemical fully connected neural network (BFCNN) that has the potential to behave like a neural network independently in vivo. In Part I, BFCNN is designed according to five different modules with concrete theoretical support on the stability of every module. In this article, we establish a systematic framework for error analysis of the designed BFCNN. We introduce current error evaluating error generation within an individual computational module by measuring the deviation between finite-time concentration and steady-state concentration, and the accumulative error characterizing error propagation across different modules connected via oscillatory signals and within the repeated modules due to multiple iterations. We further derive the formula for the total error upper bound about the iteration number and illustrate its exponential convergence order about nonzero duration time of an oscillatory signal concentration. Ultimately, the numerical experiments on two classification examples are made to exhibit the change tendency of total error bounds about the nonzero duration time and iteration number.**

***Index Terms*—Biochemical fully connected network, equilibrium approximation, error analysis, error propagation, exponential convergence.**

Notations and Abbreviations

Symbol	Description.
$\mathbb{R}^n$	n-dimensional real space.
$\mathbb{R}^n_{\geq 0}$	n-dimensional non-negative real space.
$\mathbb{R}^n_{>0}$	n-dimensional positive real space.
$\mathbb{Z}^n_{\geq 0}$	n-dimensional nonnegative integer space.
$\mathscr{C}(\times;*)$	Set of continuous differentiable functions from $\times$ to $*$.
$\lVert \cdot \rVert$	1-norm, i.e., $\lVert a\rVert = \sum_{i=1}^{n} \lvert a_i\rvert$ for $a \in \mathbb{R}^n$ and $\lVert A\rVert = max_{1\leq j\leq n} \sum_{i=1}^{m} \lvert a_{ij}\rvert$ for $A \in \mathbb{R}^{m\times n}$.
$f(B)$	$(f(b_{ij})) \in \mathbb{R}^{n\times n}$ denotes $f(\cdot)$ acts on any element of B with the real value function $f(\cdot)$ and the matrix $B = (b_{ij}) \in \mathbb{R}^{n\times n}$.
$b_{\cdot j}, b_{i\cdot}$	jth column, ith row of the matrix B, respectively.
b_{ij}	(i, j)th element of B.
$\mathbb{1}, \mathbb{0}$	Dimension-suited column vector with all elements to be 1 and 0, respectively.
BCRN	Biochemical reaction network.
BFCNN	Biochemical fully connected neural network.
CRN	Chemical reaction network.
FCNN	Fully connected neural network.
GAS	Globally asymptotically stable.
GES	Globally exponentially stable.
jBCRN	Judgment biochemical reaction network.
lwsBCRN	Linear weighted sum biochemical reaction network.
MAS	Mass action system.
ngBCRN	Negative gradient biochemical reaction network.
pBCRN	Precalculation biochemical reaction network.
sigBCRN	Sigmoid activation biochemical reaction network.
TE_m	Total error in the m iteration.
TI	Tail item.
uBCRN	Update biochemical reaction network.

I. Introduction

CELLS in nature exhibit remarkable adaptability to external stimuli through sophisticated biomolecular interactions, such as gene regulatory networks and signal pathway networks.

Received 2 March 2025; revised 22 May 2025; accepted 24 May 2025. Date of publication 27 May 2025; date of current version 29 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 12320101001 and Grant 62303409, and in part by the Jiangsu Provincial Scientific Research Center of Applied Mathematics under Grant BK20233002. An earlier version of this paper was presented at the 4th IFAC Workshop on Thermodynamics Foundations of Mathematical Systems Theory, Jul. 25-27, 2022, Canada. Recommended by Senior Editor Chief A. Astolfi. [DOI: 10.1016/j.ifacol.2022.08.022]. *(Corresponding author: Chuanhou Gao.)*

Yuzhen Fan and Chuanhou Gao are with the School of Mathematical Sciences, Zhejiang University, Hangzhou 310058, China (e-mail: yuzhen_f@zju.edu.cn; gaochou@zju.edu.cn).

Xiaoyu Zhang is with the School of Mathematics, Southeast University, Nanjing 210096, China (e-mail: Xiaoyu_Z@seu.edu.cn).

Denis Dochain is with the ICTEAM, UCLouvain, Bâtiment Euler, avenue Georges Lemaître 4-6, 1348 Louvain-la-Neuve, Belgium (e-mail: denis.dochain@uclouvain.be).

Digital Object Identifier 10.1109/TAC.2025.3574364

In synthetic biology, engineering principles are applied to design biological components and construct intricate circuits that emulate life behaviors and program cells for diverse applications such as medical care, energy, and the environment [1]. Importantly, recent progress in this field emphasizes a "system level" design, facilitating the creation of sufficiently elaborate biochemical reaction circuits to perform a wider range of tasks, such as robustness enhancement [2], [3], [4], information processing [5], [6], [7], and more.

Molecular computational modules are crucial components for performing computation and information processing within engineered biological systems. To design biochemical circuits for various computational purposes, CRNs (see Nomenclature for more abbreviations and notations), are a frequently-used theoretical tool, with dynamical equations in the form of polynomial ordinary differential equations used to model biomolecular interactions. This stems from the fact that DNA strand displacement reactions can compile any computation realizable by abstract mass-action CRNs [8]. Turing completeness of CRNs [9] also asserts this point. Significant strides have recently been made toward the goal of creating programmable biological systems to execute intricate information processing tasks, including using rate-independent reaction networks to realize feedforward neural networks based on rectified linear unit (ReLU) activation function [10], [11], [12], using phosphorylation reactions to realize multilayer perceptrons based on Sigmoid activation function [13], using reaction networks to realize supervised learning [14], [15], a smooth ReLU function with rigorous mathematical support [16], a nonlinear perceptron [17], and a linear neural network [18]. These advancements have been realized by programming artificial neural networks using BCRN systems. By leveraging the established computational framework and training mechanism of artificial neural networks, this approach enables biological circuits to solve complicated problems such as classification and decision-making efficiently and to exhibit intelligent learning ability. Nevertheless, challenges remain due to incomplete realization and theoretical limitations [19]. Specifically, these issues include the absence of assignment and judgment modules, which prevents existing biochemical neural networks from operating automatically in vivo as an artificial neural network does on a computer, as well as the lack of dynamical analysis and characterization of the inherent errors arising from steady-state-based implementation (the focus of this article).

In line with these studies, the paired article aims to automatically implement the function of a FCNN with a three-layer structure of two (input layer)-two (hidden layer)-one (output layer) nodes using BCRN, with a potential application in vivo in the future. Our primary purpose is not to develop a novel neural network that matches the accuracy and computational efficiency of other artificial neural networks in the computer environment, but rather to construct biochemical information processing circuits with learning capability in vivo serving for synthetic biology. While the current realization focuses on theoretical feasibility rather than large-scale applications, it establishes a framework for interfacing computer operations and continuous biochemical dynamics. In Part I [19], we designed a BFCNN according to assignment, feedforward propagation, judgment, learning, and clear-out modules and conducted the convergence analysis of each module. The results are shown in Table I. Under appropriate initial conditions, BFCNN behaves as a neural network to perform decision-making tasks through the evolution of its dynamics. When constructing the involved molecular computational systems, which are expected to achieve asymptotic stability, we use the limiting steady state (LSS)[1] of a certain species to represent the corresponding calculation result. Since the LSS is evaluated as time tends to be infinite while practical computation requires finite time, we have to use the species concentration evaluated at a finite time to approximate the corresponding LSS, and further to represent the practical computation result. Meanwhile, the phase length of the used chemical oscillator [19] imposes a clear limit to the operation time of each module, implying the necessity of such approximations. As a result, it inevitably produces deviations between actual concentrations and the desired calculation values. In addition, the cascade link of multiple biochemical computational modules and multiple iterations in the training process lead to the propagation of approximation errors within each module. One issue is noted [18] whether the error compromises the performance of the designed BFCNN, especially compromising the convergence of the training algorithm used in neural networks. To address these issues, an in-depth error analysis of the BFCNN is needed. The challenges lie in defining the types of error components and propagation pathways and determining the dependence of these errors on the iteration number during the training process.

In this article, we build a framework of error analysis for the designed BFCNN in Part I [19]. The main contributions are as follows.

1) We define the total error, which includes two types of error components, i.e., the *current error* to describe the error generation in each reaction computational module, and the *accumulative error* to characterize the error propagation through parameters and initial values.
2) We specify the *vertical propagation* pathway of error from the upstream module to the downstream module, and the *horizontal propagation* from previous iterations. Then, we derive the upper bound of vertical cumulative error on the weight species in the first iteration.
3) Based on the vertical cumulative error, we give the upper bound formula for the total error at the mth $(m \geq 2)$ iteration during the weight updating process. Moreover, we demonstrate that the total error is exponentially convergent concerning the finite phase length, thereby ensuring that BFCNN training inherits the convergence of the original neural network algorithm.

The rest of this article is organized as follows. Section II covers preliminaries about CRN and FCNN. The composition of error originated from the individual computational module and propagated within BFCNN are discussed in Section III. Section IV considers the error accumulated during the first iteration to illustrate impacts from individual modules and vertical propagation. Section V investigates the horizontal propagation of errors by incorporating influence from previous iterations,

[1] The *LSS* [16] of species X is defined by $\bar{x} = \lim_{t\to\infty} x(t)$ with x to denote the concentration of X.

TABLE I
ALL BIOCHEMICAL REACTION MODULES OF THE BFCNN, THE RELATED DYNAMICS AND STABILITY TYPES

Modules	Reactions	Stability & Dynamical Equations
Assignment Module	$\mathcal{O}_1$: $C_i^l + X_q^i \xrightarrow{1} C_i^l + X_q^i + S_q^l,\ S_q^l \xrightarrow{1} \varnothing,\ q \in \{1,2\},\ l \in \{1,...,\tilde{p}\}$ $C_i^l + D^i \xrightarrow{1} C_i^l + D^i + S_3^l,\ S_3^l \xrightarrow{1} \varnothing,\ i \in \{1,...,p\}$	GES $\dot{s}^l = \mathcal{X}c^l - s^l,\ s^l \in \mathbb{R}^{\tilde{p}\times 1},\ c^l \in \mathbb{R}^{p\times 1}$
	$\mathcal{O}_3$: $C_i^l \xrightarrow{k} \tilde{C}_i^l\ ,\ i \in \{1,...,p\}$	GES_rs [1] $\dot{c}^l = -kc^l,\ \dot{\tilde{c}}^l = kc^l,\ \tilde{c}^l \in \mathbb{R}^{1\times p}$
	$\mathcal{O}_5$: $\tilde{C}_{o_l}^l \xrightarrow{k} C_{\tilde{p}+o_l}^l,\ \tilde{C}_{p-\tilde{p}+l}^l \xrightarrow{k} C_l^l,\ \tilde{C}_{j_l}^l \xrightarrow{k} C_{j_l}^l$	GES_rs $\dot{c}_l^l = k\tilde{c}_{p-\tilde{p}+l}^l,\ \dot{c}_{o_l+\tilde{p}}^l = k\tilde{c}_{o_l}^l,$ $\dot{c}_{j_l}^l = k\tilde{c}_{j_l}^l,\ \ \dot{\tilde{c}}_i^l = -k\tilde{c}_i^l,\ i \in \{1,...,p\}$
Feedforward Module *lwsBCRN*	$\mathcal{O}_7/\mathcal{O}_{15}$: [2] $W_i^{\pm} + S_1^l \xrightarrow{1} W_i^{\pm} + S_1^l + N_{il}^{\pm},\ B_i^{\pm} \xrightarrow{1} B_i^{\pm} + N_{il}^{\pm}$ $W_{i+2}^{\pm} + S_2^l \xrightarrow{1} W_{i+2}^{\pm} + S_2^l + N_{il}^{\pm},\ N_{il}^{\pm} \xrightarrow{1} \varnothing, i \in \{1,2\}$	GES $\dot{n}^+ = w_1^+ s_a - n^+,\ \ \dot{n}^- = w_1^- s_a - n^-$ $n^{\pm} \in \mathbb{R}^{2\times\tilde{p}},\ w_1^{\pm} \in \mathbb{R}^{2\times 3},\ s_a \in \mathbb{R}^{3\times\tilde{p}}$
sigBCRN	$\mathcal{O}_9/\mathcal{O}_{17}$: $N_{il}^+ + N_{il}^- \xrightarrow{k} \varnothing,$	GES_rs $\dot{n}_{il}^- = \dot{n}_{il}^+ = -n_{il}^+ n_{il}^-,\ i \in \{1,2\}$
	$\mathcal{O}_{11}/\mathcal{O}_{19}$: $N_{il}^{\pm} + Half \xrightarrow{1} N_{il}^{\pm} + Half + P_{il}^{\pm},$ $N_{il}^{\pm} + P_{il}^{\pm} \xrightarrow{1} N_{il}^{\pm}$	GES $\dot{p}_{il}^{\pm} = n_{il}^{\pm}(0.5 - p_{il}^{\pm}),\ i \in \{1,2\}$
	$\mathcal{O}_{13}/\mathcal{O}_{21}$: $p_{il}^+(0) > 0 :\ N_{il}^+ + P_{il}^+ \underset{1}{\overset{1}{\rightleftarrows}} 2P_{il}^+ + N_{il}^+\ ,\ \ N_{il}^+ \xrightarrow{1} \varnothing$ $p_{il}^-(0) > 0 :\ N_{il}^- + 2P_{il}^- \xrightarrow{1} 3P_{il}^- + N_{il}^-,$ $N_{il}^- + P_{il}^- \xrightarrow{1} N_{il}^- \xrightarrow{1} \varnothing$ $P_{il}^+ \xrightarrow{1} P_{il}^+ + P_i^l,\ P_{il}^- \xrightarrow{1} P_{il}^- + P_i^l,\ P_i^l \xrightarrow{1} \varnothing$	GAS_es $\dot{p}_{il}^+ = n_{il}^+ p_{il}^+(1 - p_{il}^+),\ \dot{n}_{il}^+ = -n_{il}^+,$ $\dot{p}_{il}^- = n_{il}^- p_{il}^-(p_{il}^- - 1),\ \dot{n}_{il}^- = -n_{il}^-,$ $\dot{p}_{il} = p_{il}^+ + p_{il}^- - p_{il}.$
Judgment Module *pBCRN*	$\mathcal{O}_{23}$: $Y^l \xrightarrow{k} Y_e^l + Y_s^l + \tilde{Y}^l,\ Y_e^l + S_3^l \xrightarrow{k} \varnothing,\ Y_s^l + I_y^l \xrightarrow{k} \varnothing$ $P_i^l \xrightarrow{k} P_{is}^l + \tilde{P}_i^l,\ P_{is}^l + I_{p_i}^l \xrightarrow{k} \varnothing,\ \ i \in \{1,2\}$ $S_3^l \xrightarrow{1} E_l^+ + S_3^l,\ \ E_l^+ \xrightarrow{1} \varnothing,\ \ Y_e^l \xrightarrow{1} E_l^- + Y_e^l,\ \ E_l^- \xrightarrow{1} \varnothing$ $I_{p_i}^l \xrightarrow{1} S_{p_i}^l + I_{p_i}^l,\ \ S_{p_i}^l \xrightarrow{1} \varnothing,\ \ i \in \{1,2\}$ $I_y^l \xrightarrow{1} S_y^l + I_y^l,\ \ S_y^l \xrightarrow{1} \varnothing$ $E_l^+ \xrightarrow{1} E_l^+ + E_l,\ E_l^- \xrightarrow{1} E_l^- + E_l,\ E_l \xrightarrow{1} \varnothing$	Finite-time exponential stability $\dot{y}^l = -ky^l,\ \dot{\tilde{y}}^l = ky^l,\ \dot{y}_e^l = ky^l - ky_e^l s_3^l$ $\dot{p}_{il} = -kp_{il},\ \dot{\tilde{p}}_{il} = kp_{il},\ \dot{p}_{is}^l = kp_{il} - kp_{is}^l i_{p_i}^l,\ \ \dot{i}_{p_i}^l = -kp_{is}^l i_{p_i}^l$ $\dot{s}_3^l = -ky_e^l s_3^l,\ \ \dot{y}_s^l = ky^l - ky_s^l i_y^l,\ \ \dot{i}_y^l = -ky_s^l i_y^l$ $\dot{e}_l^+ = s_3^l - e_l^+,\ \dot{e}_l^- = y_e^l - e_l^-,\ \ \dot{s}_y^l = i_y^l - s_y^l,\ \ \dot{e}_l = e_l^+ + e_l^- - e_l$
jBCRN	$\mathcal{O}_{25}$: $A_l \xrightarrow{k_1} 2E_l,\ \ 2E_l \xrightarrow{k_2} E_l + A_l,$ $E_l + A_l \xrightarrow{k_3} A_l,\ \ E_l \xrightarrow{k_4} \varnothing$ $E_l \xrightarrow{k} E_l + C_a,\ C_a \xrightarrow{k} \varnothing,\ \forall l \in \{1,...,\tilde{p}\}$	Bistability $\dot{e}_l = 2k_1 a_l - k_2 e_l^2 - k_3 e_l a_l - k_4 e_l$ $\dot{a}_l = -k_1 a_l + k_2 e_l^2,\ \ \dot{c}_a = ke_l - kc_a$
Learning Module *ngBCN* (e.g., 6-factor CRN for multiplication + the negative part of gradient for addition)	$\mathcal{O}_{27}$: *Multiplication* $X_1 + X_2 \xrightarrow{1} X_1 + X_2 + Z_1,\ Z_1 \xrightarrow{1} \varnothing$ $X_3 + X_4 \xrightarrow{1} X_3 + X_4 + Z_2,\ Z_2 \xrightarrow{1} \varnothing$ $X_5 + X_6 \xrightarrow{1} X_5 + X_6 + Z_3,\ Z_3 \xrightarrow{1} \varnothing$ $Z_1 + Z_2 \xrightarrow{1} Z_1 + Z_2 + Y_1,\ \ \ Y_1 \xrightarrow{1} \varnothing$ $Z_3 + Y_1 \xrightarrow{1} Z_3 + Y_1 + Q,\ \ \ Q \xrightarrow{1} \varnothing$ *Addition* $Q_{s+-}^l \xrightarrow{1} Q_{s+-}^l + Par_s^-$ $Q_{s-+}^l \xrightarrow{1} Q_{s-+}^l + Par_s^-$ $Q_{v-}^l \xrightarrow{1} Q_{v-}^l + Par_v^-$ $Par_s^- \xrightarrow{1} \varnothing$ [1] $Par_v^- \xrightarrow{1} \varnothing$ [1]	GES $\dot{z}_1 = x_1 x_2 - z_1,\ \ \ \dot{z}_2 = x_3 x_4 - z_2$ $\dot{z}_3 = x_5 x_6 - z_3,\ \ \ \dot{y}_1 = z_1 z_2 - y_1,\ \ \ \dot{q} = z_3 y_1 - q$ $\dot{par}_v^- = \sum_{l=1}^{\tilde{p}} q_{v-}^l - par_v^-\ ,\ \ \dot{par}_s^- = \sum_{l=1}^{\tilde{p}} (q_{s+-}^l + q_{s-+}^l) - par_s^-$
uBCRN	$\mathcal{O}_{29}$: $Par_i^{\pm} + L \xrightarrow{1} Par_i^{\pm} + L + \Delta W_i^{\pm},\ \Delta W_i^{\pm} \xrightarrow{1} W_i^{\pm},$ $\Gamma_i^{\pm} \xrightarrow{1} \Gamma_i^{\pm} + W_i^{\pm},\ W_i^{\pm} \xrightarrow{1} \varnothing,$ $Par_{j+6}^{\pm} + L \xrightarrow{1} Par_{j+6}^{\pm} + L + \Delta B_j^{\pm},\ \Delta B_j^{\pm} \xrightarrow{1} B_j^{\pm},$ $\Gamma_{j+6}^{\pm} \xrightarrow{1} \Gamma_{j+6}^{\pm} + B_j^{\pm},\ B_j^{\pm} \xrightarrow{1} \varnothing$	GES $\Delta\dot{w}^{\pm} = \eta par^{\pm} - \Delta w^{\pm},\ \dot{w}^{\pm} = \gamma^{\pm} + \Delta w^{\pm} - w^{\pm}$ $\Delta w^{\pm},\ par^{\pm},\ w^{\pm} \in \mathbb{R}^{3\times 3},\ i \in \{1,...,6\}, j \in \{1,2,3\}$
Clear-out Module	$\mathcal{O}_{31}$: *clear − out BCRN* $X \xrightarrow{1} \varnothing$ *compensatory BCRN* $I \xrightarrow{1} I + Y$ $Y \xrightarrow{1} \varnothing$	GES $\dot{x} = -x$ $\dot{y} = 1 - y$

[1] GES means globally exponentially stable; GES_rs means GES relative to its stoichiometry compatibility class and GAS_es means GAS convergence to the equilibrium set. For the subscripts in the assignment module, we have $o_l \in I_l,\ j_l \notin I_l \bigcup \{p - \tilde{p} + l\}$ with $I_l \triangleq \{l, \tilde{p} + l, ..., p - 2\tilde{p} + l\}$ for any fixed l. In particular, $j_l \in \varnothing$ if $\tilde{p} = 1$ implying that $\tilde{C}_{j_l}^l \xrightarrow{k} C_{j_l}^l$ in $\mathcal{O}_5$ are absent, and $I_l \triangleq \{l\}$ if $3\tilde{p} > p$, i.e., $\tilde{p} + l > p - 2\tilde{p} + l$. Besides, $l \in \{1, ..., \tilde{p}\}$, $s \in \{1,2,3,4,7,8\}$, $v \in \{5,6,9\}$ and other subscripts are specified in this table.

[2] In the Feedforward Module, the two-phase mark, i.e., $\mathcal{O}_i/\mathcal{O}_j$, implies that the type of BCRN listed in the considering row implements the linear/nonlinear feedforward computation in the input-hidden layer (operate in the smaller-subscript phase) and hidden-output layer (operate in the bigger-subscript phase), respectively.

thereby presenting the total error upper bound formula for any iteration number. The change trends of the upper bound concerning the phase length and iteration number are validated numerically in Section VI. Finally, Section VII concludes this article.

II. PRELIMINARIES

In this section, we formally introduce the basic concepts of CRNs [20], [21], [22] and present an overview of FCNN with the specific feedforward structure of an input layer, a hidden layer, and an output layer. In this paired article, the former is used to program the latter. Specifically, in Part I, a BFCNN is constructed for the case that the FCNN has a three-layer structure with two (input layer)-two (hidden layer)-one (output layer) nodes.

A. Chemical Reaction Network

Consider a CRN with n species that interact with each other through r reactions.

Definition 1 (CRN): A CRN consists of three finite sets, i.e., *species set:* $\mathcal{S} = \{X_1, \ldots, X_n\}$ denotes the subjects participated in reactions; *complex set:* $\mathcal{C} = \bigcup_{j=1}^{r} \{C_j, C'_j\}$ whose elements have the form $\sum_{i=1}^{n} a_i X_i$ with $a \in \mathbb{Z}_{\geq 0}^{n}$ called the complex vector and its entry a_i called the stoichiometric coefficient of X_i; $\mathcal{R} = \bigcup_{j=1}^{r}\{C_j \to C'_j\}$ describe interactions between species, often referred to as a triple $(\mathcal{S}, \mathcal{C}, \mathcal{R})$.

Based on the above definition, the jth reaction is written as

$$\sum_{i=1}^{n} v_{ij} X_i \longrightarrow \sum_{i=1}^{n} v'_{ij} X_i \tag{1}$$

where $(v_{1j}, \ldots, v_{nj})^\top = v_{\cdot j}, (v'_{1j}, \ldots, v'_{nj})^\top = v'_{\cdot j}$ are called *reactant complex vector* and *product complex vector*, respectively. The dynamics of $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ that captures the change of the concentration of each species, labeled by $x \in \mathbb{R}^n_{\geq 0}$, may be reached if a r-dimensional vector-valued function $\mathscr{K} \in \mathscr{C}(\mathbb{R}^n; \mathbb{R}^r)$ is defined to evaluate the reaction rates and the balance law is further utilized, written as

$$\frac{dx(t)}{dt} = \Gamma \mathscr{K}(x), \ x \in \mathbb{R}^n_{\geq 0} \tag{2}$$

where $\Gamma \in \mathbb{Z}^{n \times r}$ is the *stoichiometric matrix* with the jth column $\Gamma_{\cdot j} = v'_{\cdot j} - v_{\cdot j}$. Usually, *mass action kinetics* is used to characterize the reaction rate, which induces the rate of the jth reaction as $\mathscr{K}_j(x) = k_j x^{v_{\cdot j}} \triangleq k_j \prod_{i=1}^{n} x_i^{v_{ij}}$, where $k_j > 0$ represents the rate constant. The CRN system endowed with mass action kinetics offers polynomial ordinary differential equations, often termed as MAS and denoted by the quad $(\mathcal{S}, \mathcal{C}, \mathcal{R}, k)$ with $k = (k_1, \ldots, k_r)^\top$.

B. Fully Connected Neural Network

Assume a FCNN to perform a classification task for the input–output dataset $\mathbb{D} = \{(\tilde{x}^i, d^i)\}_{i=1}^{p}$ with p to represent the size of $\mathbb{D}$ and d^i to indicate the label of $\tilde{x}^i$. The training process utilizes the common stochastic gradient descent algorithm with the mini-batch strategy (i.e., at each iteration only part of samples are fed to update weights), and the activation function f takes the *Sigmoid function*. For simplicity but without the loss of generality (that means the techniques developed here can be easily extended to other more complicated neural networks), we fix the structure of FCNN to be three-layers with *two (input layer)-two (hidden layer)-one (output layer)* nodes to exhibit this process.

1) Feedforward Propagation: Denote the sample matrix by $\chi \in \mathbb{R}^{3 \times p}$ with the ith $(i = 1, \ldots, p)$ column $\chi_{.i} = (\tilde{x}_1^i, \tilde{x}_2^i, d^i)^\top$, and the *input matrix* by $\Xi \in \mathbb{R}^{3 \times \tilde{p}}_{\geq 0}$ with the mini-batch size $\tilde{p}$ $(\tilde{p} \leq p)$ and the lth column $\Xi_{.l} = (\tilde{x}_1^l, \tilde{x}_2^l, 1)^\top$ $(l = 1, \ldots, \tilde{p})$. Note that the third row of Ξ storing constant 1 serves to handle the bias in the input layer. Further, let the *weight matrices* connecting input-hidden layers and hidden-output layers be $\mathcal{W}_1 \in \mathbb{R}^{2 \times 3}$ and $\mathcal{W}_2 \in \mathbb{R}^{1 \times 3}$, respectively, and their last columns also represent the corresponding bias. Then, the feedforward computation process at one iteration follows

$$\mathcal{N} = \mathcal{W}_1 \cdot \Xi, \Upsilon = f(\mathcal{N}), \tilde{\Upsilon} = \begin{pmatrix} \Upsilon \\ \mathbb{1}^\top \end{pmatrix}$$
$$\tilde{\mathcal{N}} = \mathcal{W}_2 \cdot \tilde{\Upsilon}, y = f(\tilde{\mathcal{N}}) \tag{3}$$

where $\mathcal{N} \in \mathbb{R}^{2 \times \tilde{p}}$ stores the linear weighted sums, $\Upsilon \in \mathbb{R}^{2 \times \tilde{p}}$ is called the *hidden matrix* storing the results of the Sigmoid activation function f on $\mathcal{N}$, $\tilde{\Upsilon} \in \mathbb{R}^{3 \times \tilde{p}}$ is constructed to add a row of 1 to Υ to handle bias in the hidden layer, $\tilde{\mathcal{N}} \in \mathbb{R}^{1 \times \tilde{p}}$, and $y \in (0,1)^{1 \times \tilde{p}}$ is the *output matrix*. We further can compute the training loss $\mathcal{E}(\mathcal{W}_1, \mathcal{W}_2) = \frac{1}{2}(\delta - y)(\delta - y)^\top$, where $\delta = (d^1, \ldots, d^{\tilde{p}})$. This finishes the feedforward computation of one iteration with $\tilde{p}$ samples. Note that for the classification task, the prediction label takes values in $\{0, 1\}^{1 \times \tilde{p}}$ with label $= 1$ if $y^i > \frac{1}{2}$ and label $= 0$ otherwise. If the regression task is faced, the prediction output will be assigned with the actual value of $y^i = f^i(\tilde{\mathcal{N}})$ for the whole training.

2) Backward Propagation: This process serves to update weights from the mth iteration to the $(m+1)$th iteration according to the stochastic gradient descent algorithm, which takes

$$\mathcal{W}^{m+1} = \mathcal{W}^m + \Delta \mathcal{W}^m \tag{4}$$

with $\mathcal{W} = \binom{\mathcal{W}_1}{\mathcal{W}_2}$ and the (i, j)th element of $\Delta \mathcal{W}^m$ to be

$$\Delta \mathcal{W}^m_{1_{ij}} = -\eta \frac{\partial \mathcal{E}^m}{\partial \mathcal{W}^m_{1_{ij}}} = -\eta \sum_{l=1}^{\tilde{p}} e_l^m f'(n_{3l}^m) \mathcal{W}^m_{2_{1i}} f'(n_{il}^m) \Xi^m_{jl}$$
$$\Delta \mathcal{W}^m_{2_{1j}} = -\eta \frac{\partial \mathcal{E}^m}{\partial \mathcal{W}^m_{2_{1j}}} = -\eta \sum_{l=1}^{\tilde{p}} e_l^m f'(n_{3l}^m) \tilde{\Upsilon}^m_{jl}$$
$$i = 1, 2; \ j = 1, 2, 3. \tag{5}$$

Here, $\eta \in (0, 1]$ is the learning rate, $e_l^m = (\delta^m - y^m)_{1\,l}$, $(n_{1\,l}^m, n_{2\,l}^m)^\top = \mathcal{N}^m_{\cdot l}$, $n_{3\,l}^m = \tilde{\mathcal{N}}^m_{1\,l}$, and the differentiation is derived by chain rule. Then, the training is terminated if $|e_l^m|$ is less than the preset threshold for all l, otherwise, it continues with another group of $\tilde{p}$ samples from $\mathbb{D}$.

In the subsequent error analysis, each computational value (e.g. $\mathcal{N}_{il}, y$, etc.) in the FCNN is referred to as the desired value, uniformly represented by symbol x^* for convenience. Namely, x^* is designed as the desired LSS of the corresponding species in BFCNN.

III. Composition of Errors

This section provides a comprehensive overview of the error composition throughout the operation process. According to Table I, the BFCNN encompasses five primary modules, i.e., the assignment, feedforward, judgment, learning, and clear-out modules, each incorporating several computational reaction subsystems. The chemical oscillator [23] in the form of

$$O_1 + O_2 \xrightarrow{k_o} 2O_2, \ O_2 + O_3 \xrightarrow{k_o} 2O_3,$$
$$\ldots\ldots \quad , \ O_r + O_1 \xrightarrow{k_o} 2O_1 \tag{6}$$

governs the sequential execution of computational reaction subsystems by assigning distinct odd-encoded oscillatory species O_i as catalysts for different systems. It implies that reactions within each module occur exclusively during the nonzero phase of their respective catalysts O_j, denoted as $\mathcal{O}_j$ in Table I. For example, for the reactions $C^l_{i_2} \xrightarrow{k} \tilde{C}^l_{i_2}$ driven by the phase $\mathcal{O}_3$ in the assignment module, they actually occur according to $C^l_{i_2} + O_3 \xrightarrow{k} \tilde{C}^l_{i_2} + O_3$. Apparently, when the concentration

TABLE II
TOTAL ERRORS NEEDED TO BE ESTIMATED ON THE UPPER BOUNDS FOR THE BFCNN

Modules	Error	Description
Assignment Module	$\mathcal{O}_1 : \|s^m(T) - s^{*,m}\|$	TE_m of input species
	$\mathcal{O}_5 : \|c^m(3T) - c^{*,m}\|,\ \|\tilde{c}^m(2T) - \tilde{c}^{*,m}\|$	TE_m of order species, auxiliary order species
Feedforward Module	$\mathcal{O}_7/\mathcal{O}_{15} : \lvert n_{il}^{\pm,m}(T) - n_{il}^{\pm,*,m}\rvert,\ \|\tilde{n}^{\pm,m}(T) - \tilde{n}^{\pm,*,m}\|$	TE_m of positive/negative net input species
	$\mathcal{O}_{13} : \lvert p_{il}^m(T) - p_{il}^{*,m}\rvert\ /\ \|p^m(T) - p^{*,m}\|$	TE_m of hidden-layer output species
	$\mathcal{O}_{21} : \lvert y_l^m(T) - y_l^{*,m}\rvert\ /\ \|y^m(T) - y^{*,m}\|$	TE_m of global output species
Judgment Module	$\mathcal{O}_{23} : \lvert e_l^{\pm,m}(T) - e_l^{\pm,*,m}\rvert$	TE_m of positive/negative error species
	$\lvert\tilde{y}_l^m(T) - \tilde{y}_l^{*,m}\rvert,\ \lvert s_y^{l,m}(T) - s_y^{l,*,m}\rvert$	TE_m of reserving $y^m, 1 - y^m$
	$\lvert\tilde{p}_{il}^m(T) - \tilde{p}_{il}^{*,m}\rvert,\ \lvert s_{p_i}^{l,m}(T) - s_{p_i}^{l,*,m}\rvert$	TE_m of reserving $\Upsilon_{il}^m, 1 - \Upsilon_{il}^m$
Learning Module	$\mathcal{O}_{27} : \lvert par_{1_{ij}}^{\pm,m}(T) - par_{1,\pm_{ij}}^{*,m}\rvert$	TE_m of gradient species for the hidden layer
	$\lvert par_{2_{1j}}^{\pm,m}(T) - par_{2,\pm_{1j}}^{*,m}\rvert$	TE_m of gradient species for the output layer
	$\mathcal{O}_{29} : \|w_1^{\pm,m}(T) - w_{\pm,1}^{*,m}\|$	TE_m of hidden-layer weight species
	$\|w_2^{\pm,m}(T) - w_{\pm,2}^{*,m}\|$	TE_m of output-layer weight species
Clear-out Module	$\mathcal{O}_{31} : \lvert x^m(\beta T)\rvert$	TE_m of species X to be cleared out
	$\lvert y^m(\beta T) - 1\rvert$	TE_m of species Y to be compensated

TABLE III
PHASE LENGTH OF (6) WITH $r = 32$, $k_o = 1$, $tspan = [0\,900]$, AND THE LOW LEVEL 5×10^{-6}

	$\mathcal{O}_1$	$\mathcal{O}_3$	$\mathcal{O}_5$	$\mathcal{O}_7$	$\mathcal{O}_9$	$\cdots$
Cycle 1	20.1613	20.2588	20.1751	20.0439	20.3423	$\cdots$
Cycle 2	20.2947	20.2294	20.2852	20.2582	20.1754	$\cdots$
Cycle 3	20.0626	20.2401	20.1389	20.2844	20.3260	$\cdots$
Cycle 4	20.2392	20.2545	20.1977	20.1612	20.2149	$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

of O_3 is nonzero, these reactions can occur. Moreover, in the phase $\mathcal{O}_3$, other odd-encoded oscillatory species are all almost zeroes. Note that the nonzero duration time of each phase, called the *phase length*, in the oscillating system (6) is the same, and we label it by $T \in (0, \infty)$ for convenience.

Remark 1: To specify the assumption of the phase length, we consider (6) consisting of three species [23], [24] and given initial concentrations $[10^{-6}, 10^{-6}, 1]$ and $k_o = 1$. Its trajectory lies in a fixed periodic orbit with a fixed period $\tilde{T}$ and amplitude. The phase length, in practice, refers to the duration time of the concentration exceeding a very low level, e.g., 10^{-6} here. For species O_1, two points $o_1(t_1) = o_1(t_2) = 10^{-6}$, where in the next time step $o_1(t_1)$ leading to an increase and $o(t_2)$ to a decrease, can be found on this periodic orbit. If $o_1(t_1)$ and $o_1(t_2)$ are within the same period, the phase length $T = |t_2 - t_1| < \tilde{T}$ is invariant due to the property of the periodic orbit. Moreover, since each species is symmetric, $|t_2 - t_1|$ is the same for each species. Therefore, it is reasonable to assume that given k_o and initial concentrations of (6), the actual reaction time of computational subsystems controlled by oscillatory species is fixed at T. In addition, for $r > 3$, we present the numerical results of the first few phases and periods in Table III using the oscillator employed by BFCNN. The slight differences may arise from numerical errors.

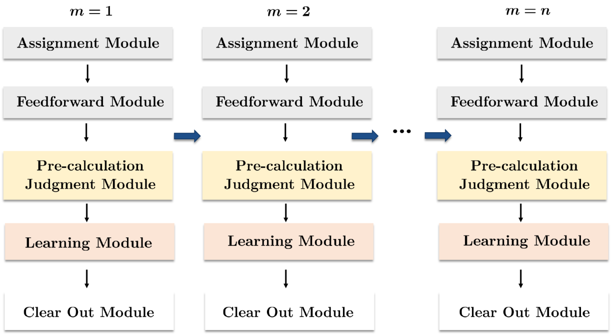

Fig. 1. Propagation of the current error. The vertical thin black arrow denotes the vertical propagation, and the horizontal thick arrow describes the horizontal propagation.

As discussed in Section I, in each reaction subsystem of the modules listed in Table I, an error arises when using the species concentration evaluated at a finite time to approximate its current LSS, called *Current error* here. This error propagates along both vertical (from the upstream module to the downstream module) and horizontal (from the previous iteration to the present iteration) pathways, as displayed in Fig. 1. The propagation error is called *Accumulative error* in the context, evaluated by the difference between the current LSS and the desired LSS. Their definitions are provided below.

1) Current Error:

Consider species X, which may participate in multiple computational reaction subsystems driven by different oscillating signals, and denote a representative subsystem by $\mathcal{M}_j$. The current error for X in $\mathcal{M}_j$ is defined by $|x(\beta T) - \bar{x}| < \bar{\epsilon}_j$, where $\beta \in \mathbb{Z}_{>0}$ is the number of the reaction subsystems that X participates in, and moreover have already taken place up to now. βT records the current time and $\bar{\epsilon}_j$ is the current error bound. Note that the above $\bar{x}$ represents the current LSS of X and should be different in different reaction subsystems.

2) Accumulative Error: The *Current error* implies that $x(\beta T)$ will replace $\bar{x}$ to be used for the next reaction subsystem $\mathcal{M}'_j$ (after $\mathcal{M}_j$) that X participates in. The caused error will further propagate along two pathways: vertically from upstream modules to the present module, and horizontally from previous iterations to the present iteration, i.e., error accumulation happens. Since $x(\beta T)$ instead of $\bar{x}$ in $\mathcal{M}_j$ acts as a new initial value entering into $\mathcal{M}'_j$, it may lead to the LSS of X in $\mathcal{M}'_j$ to deviate from its desired value, which is generated based on the design of the BFCNN. We thus describe the *Accumulative error* in $\mathcal{M}_j$ as $|\bar{x} - x^*| < \tilde{\epsilon}_j$ with x^* to be the desired LSS of X in $\mathcal{M}_j$ and $\tilde{\epsilon}_j$ to be the bound.

Combining these two kinds of errors, we can estimate the upper bound for the total deviation of the output concentration of X in $\mathcal{M}_j$ to be

$$\begin{aligned} |x(\beta T) - x^*| &< |x(\beta T) - \bar{x}| + |\bar{x} - x^*| \\ &< \bar{\epsilon}_j + \tilde{\epsilon}_j \triangleq \epsilon_j. \end{aligned} \tag{7}$$

Table II summarizes all of the total errors (TE_m) that need to be estimated on the bounds for the BFCNN according to different modules. There, we use the mark "m" in the superscript to represent the mth iteration.

In the following, we take the weight update process of (4) and (5) as an example to estimate the corresponding TE_m. Consider the weight matrix $\mathcal{W}^m$ in (4), in which since each element needs to be encoded by the difference of two nonnegative numbers, we write $\mathcal{W}^m = \mathcal{W}^m_+ - \mathcal{W}^m_-$. Here, $\mathcal{W}^m_+/\mathcal{W}^m_- \in \mathbb{R}^{3\times 3}_{\geq 0}$ is constructed by the first/second nonnegative numbers, and is called the positive/negative part of $\mathcal{W}^m$. We express them as $\mathcal{W}^m_\pm$ for simplicity in the context, which also applies to other ± marks. As shown in the "*uBCRN*" row of Table I, the weight species concentration matrix w^m is used to program $\mathcal{W}^m$, by referencing (7) we thus estimate its TE_m (only the positive part is shown) to be

$$\begin{aligned} &\|w^{+,m}(T) - \mathcal{W}^{m+1}_+\| \\ &< \bar{\epsilon}^m_{w_+} + \|w^{+,m-1}(T) - \mathcal{W}^m_+\| + \eta\theta^m_+ \\ &< \bar{\epsilon}^m_{w_+} + \epsilon^{m-1}_{w_+} + \eta\theta^m_+ \end{aligned} \tag{8}$$

where $\theta^m_+ = \|\nabla\mathcal{E}^m_+(w^{\pm,m-1}(T)) - \nabla\mathcal{E}^m_+(\mathcal{W}^m_\pm)\|$ originates from computing the gradient of $\mathcal{E}^m_+$ in the mth iteration. Note that here $w^{+,m}(\beta T) = w^{+,m}(T)$, i.e., $\beta = 1$, since the reactions in the uBCRN are driven by the only oscillatory signal O_{29}. The above inequality implies that we can take $\tilde{\epsilon}^m_{w_+} = \epsilon^{m-1}_{w_+} + \eta\theta^m_+$. Then, this yields the iteration formula for *Accumulative error*

$$\tilde{\epsilon}^m_{w_+} = \tilde{\epsilon}^{m-1}_{w_+} + \bar{\epsilon}^{m-1}_{w_+} + \eta\theta^m_+.$$

Further, we obtain a rough expression for TE_ m bound

$$\epsilon^m_{w+} = \bar{\epsilon}^m_{w+} + \tilde{\epsilon}^m_{w_+} = \bar{\epsilon}^m_{w+} + \sum_{j=0}^{m-1}(\bar{\epsilon}^j_{w_+} + \eta\theta^{j+1}_+) \tag{9}$$

where $\bar{\epsilon}^0_{w_+} = 0$ since initial weight concentrations are injected in advance and not subject to calculation. In (9), $\bar{\epsilon}^j_{w_+} + \eta\theta^{j+1}_+$ can be interpreted as the error accumulated vertically from all upstream modules up to the uBCRN system at $j+1$th iteration. The summation from zero to $m-1$ shows the horizontal propagation. A similar argument can be applied in deducing $\bar{\epsilon}^m_{w_-}, \tilde{\epsilon}^m_{w_-}$ and then obtaining ϵ^m_w by $\epsilon^m_{w_+} + \epsilon^m_{w_-}$.

If the training process is confirmed to be convergent, i.e., $\forall\epsilon > 0$, $\exists K \in \mathbb{Z}_{>0}$ s.t. for all $m \geq K$ we have $\|\mathcal{W}^m - \mathcal{W}^{\text{opt}}\| <\epsilon$, and the TE_m ϵ^m_w caused by BCRN dynamics is also small enough to vanish with increasing finite time T, the actual weight concentration matrix sequences can approach the accurate solution in the algorithm with any given precision. In the following sections, based on the analytical framework described above, we will proceed to estimate errors in all phases specific to the content in $\bar{\epsilon}^m$ and θ^m and show all the error upper bounds and their convergence order.

IV. Error Estimation in the First Iteration

In this section, our attention is directed toward the error vertically propagated from upstream modules to the learning module. Considering the design of BFCNN, certain species serve as computational output and input species for two adjacent modules, respectively, leading to these species carrying deviated computational results from one module to the next. Therefore, our derivation of errors cumulated along the vertical propagation pathway follows the operational sequence of the modules. To highlight the key process of error propagation, we present only the total error bound for those species that serve as computational outputs within each module and ultimately give the TE_1 bound in weight updating. Suppose that no errors arise during the injection of any species X with the initial concentration into reactors, implying that $x^1(0)$ is accurate. Hereafter, $\bar{x}^m$ and $x^{*,m}$ appearing in the specific subsection indicate the variables of the considering subsystem in the mth iteration unless otherwise specified as $\{\bar{x}^m\}_{\mathcal{O}_i}$, $\{x^{*,m}\}_{\mathcal{O}_i}$, and the mark $m = 1$ are neglected in this section for convenience. Moreover, parameters in scalar estimation vary based on the input sample index l but in the text, they are taken to be the ones of the largest item about l.

A. Assignment Module Error

As can be seen from Table I, the assignment module consists of three computational reaction subsystems $\mathcal{M}^a_1, \mathcal{M}^a_2, \mathcal{M}^a_3$ (Here, the superscript a is used to identify "assignment module". The analogous identifications are used to express other modules in the subsequent.) occupying three phases $\mathcal{O}_1, \mathcal{O}_3, \mathcal{O}_5$ of one period, and each computational reaction subsystem is the MAS demonstrated to permit GES equilibrium points relative to its stoichiometry compatibility class [19]. For $\forall i \in \{1, \ldots, p\}$, $l \in \{1, \ldots, \tilde{p}\}$, species of interest are, the sample species set $\{X^i_1, X^i_2, D^i\}$, input species set $\{S^l_1, S^l_2, S^l_3\}$, order species set $\{C^l_i\}$, and auxiliary order species set $\{\tilde{C}^l_i\}$ with their concentration vectors $\mathcal{X}_{.i}(t) = (x^i_1(t), x^i_2(t), d^i(t))^\top$, $s^l(t) = (s^l_1(t), s^l_2(t), s^l_3(t))^\top$, $c^l(t) =$

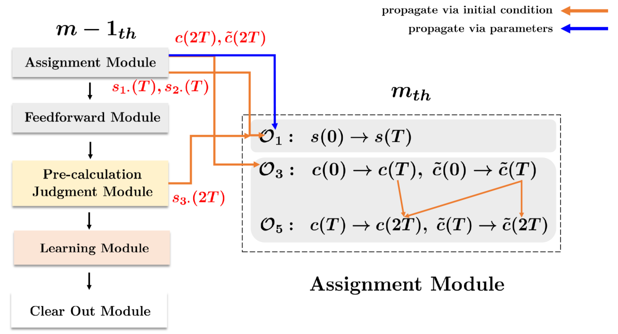
propagate via initial condition
propagate via parameters
$m-1_{th}$
$c(2T), \tilde{c}(2T)$
Assignment Module
$s_{1\cdot}(T), s_{2\cdot}(T)$
Feedforward Module
Pre-calculation Judgment Module
$s_{3\cdot}(2T)$
Learning Module
Clear Out Module
m_{th}
$\mathcal{O}_1: \ s(0) \to s(T)$
$\mathcal{O}_3: \ c(0) \to c(T), \ \tilde{c}(0) \to \tilde{c}(T)$
$\mathcal{O}_5: \ c(T) \to c(2T), \ \tilde{c}(T) \to \tilde{c}(2T)$
Assignment Module

Fig. 2. Schema of the propagation of error through initial concentrations from the $m-1$th to the mth iteration in the Assignment Module.

$(c^l_1(t),\ldots,c^l_p(t))^\top$, $\tilde{c}^l(t) = (\tilde{c}^l_1(t),\ldots,\tilde{c}^l_p(t))^\top$, respectively. Fig. 2 shows that, when $m=1$, only current errors appear in $\mathcal{M}^a_1, \mathcal{M}^a_2$, but the accumulative error in $\mathcal{M}^a_3$ caused by the vertical propagation from $\mathcal{M}^a_2$ is inevitable since $c^l(2T), \tilde{c}^l(T)$ become its initial condition. Thus, with some difference from the notation in Part I, given $s(t) \in \mathbb{R}^{3\times\tilde{p}}_{\geq 0}$, $c(t), \tilde{c}(t) \in \mathbb{R}^{p\times\tilde{p}}_{\geq 0}$ with $s^l(t), c^l(t), \tilde{c}^l(t)$ being their columns, respectively, and combining $\mathcal{X}(t) \in \mathbb{R}^{3\times p}$, we have the following results for the first iteration.

Proposition 1: For the assignment module given in Table I, let the desired values $s^* = (\chi_{\cdot 1},\ldots,\chi_{\cdot\tilde{p}})$, $\tilde{c}^* = \mathbb{0}_{p\times\tilde{p}}$, $c^* = (\xi^\top_{1+\tilde{p}},\ldots,\xi^\top_{2\tilde{p}})$, where $\xi_{l+\tilde{p}} \in \mathbb{R}^{1\times p}$ with the $l+\tilde{p}$th element being 1 and others being 0, then there are positive coefficients m_1, m_3, and $\tilde{m}_3$ such that the total error upper bounds are constrained by

$$\|s(T) - s^*\| \leq m_1 e^{-T}$$
$$\|c(3T) - c^*\| \leq m_3 e^{-kT}, \ \|\tilde{c}(2T) - \tilde{c}^*\| \leq \tilde{m}_3 e^{-kT} \quad (10)$$

where $k > 0$ is the rate constant in $\mathcal{M}^a_2$ and $\mathcal{M}^a_3$, the arguments $T, 3T$, and $2T$ mean that $\beta = 1, \beta = 3$, and $\beta = 2$, respectively (we will ignore the statement about what β is in other modules analysis).

Here, specific coefficients m_1, m_3, and $\tilde{m}_3$ are associated with training samples, k, and any choice of $s(0)$, provided in the proof in Supplementary Material (applied to other estimation parameters). The desired values s^* is for M^a_1, and $c^*, \tilde{c}^*$ are for M^a_3. Notably, in $\mathcal{M}^a_3$, $\bar{\tilde{c}} = \tilde{c}^*$ since the MAS for auxiliary order species is GES. Hence, the propagation error in initial conditions only alters the convergence characteristics, i.e., the original constant coefficient $\|\tilde{c}(T)\|$, rather than the LSS value.

B. Feedforward Module Error

The feedforward module of BFCNN comprises eight computational reaction subsystems occupying eight odd phases from $\mathcal{O}_7$ to $\mathcal{O}_{21}$, as depicted in Fig. 3, where lwsBCRN designed for modules $\mathcal{M}^f_{l_1}(\mathcal{O}_7), \mathcal{M}^f_{l_2}(\mathcal{O}_{15})$ aiming to obtain $\mathcal{N}, \tilde{\mathcal{N}}$ in (3) and nonlinear sigBCRN devised for $\mathcal{M}^f_{n^j_1}, \mathcal{M}^f_{n^j_2}$ with $j = \{1,2,3\}$ serving to compute the sigmoid activation function. Here, considering the similarity, we first focus on the deviation estimation caused by lwsBCRN and sigBCRN by taking the hidden layer as an example, i.e., $\mathcal{M}^f_{l_1}$ and $\mathcal{M}^f_{n^j_1}$. This allows us to directly show the TE_1 bound in the output layer for a single feedforward computation. Note that $n^\pm(t) \in \mathbb{R}^{2\times\tilde{p}}_{\geq 0}$ serve to denote concentrations of the net input species pair $\{N^+_{il}, N^-_{il}\}$ with $i = \{1,2\}$ identifying two nodes in the hidden layer in this section, $w^\pm_1(t) \in \mathbb{R}^{2\times 3}_{\geq 0}$, $w^\pm_2(t) \in \mathbb{R}^{1\times 3}_{\geq 0}$ are defined for concentrations of the weight species pair $\{W^+_q, W^-_q\}^6_{q=1}$ and the bias species pair $\{B^+_q, B^-_q\}^3_{q=1}$, and $p^\pm(t), p(t) \in \mathbb{R}^{2\times\tilde{p}}_{\geq 0}$ indicate concentrations of hidden-layer output species pairs $\{P^+_{il}, P^-_{il}\}$, and hidden-layer output species $\{P^l_i\}$, respectively [19].

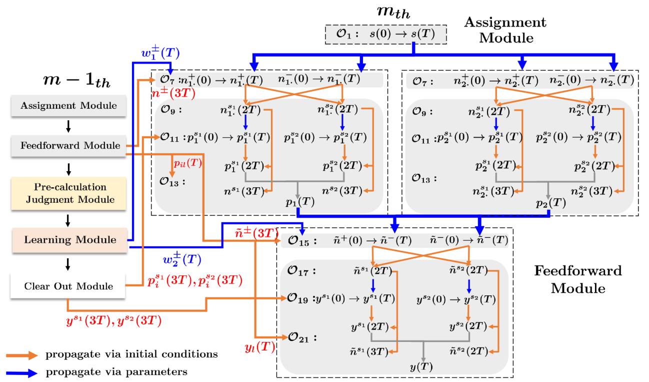
m_{th}
$\mathcal{O}_1: \ s(0) \to s(T)$
Assignment Module
$w^\pm_1(T)$
$m-1_{th}$
Assignment Module
Feedforward Module
Pre-calculation Judgment Module
Learning Module
Clear Out Module
$\mathcal{O}_7: n^+_{1\cdot}(0) \to n^+_{1\cdot}(T) \quad n^-_{1\cdot}(0) \to n^-_{1\cdot}(T)$
$n^\pm(3T)$
$\mathcal{O}_9:$
$n^{s_1}_{1\cdot}(2T)$
$n^{s_2}_{1\cdot}(2T)$
$\mathcal{O}_{11}: p^{s_1}_1(0) \to p^{s_1}_1(T) \quad p^{s_2}_1(0) \to p^{s_2}_1(T)$
$p_{il}(T)$
$p^{s_1}_1(2T)$
$p^{s_2}_1(2T)$
$\mathcal{O}_{13}:$
$n^{s_1}(3T)$
$n^{s_2}(3T)$
$p_1(T)$
$\mathcal{O}_7: \ n^+_{2\cdot}(0) \to n^+_{2\cdot}(T) \quad n^-_{2\cdot}(0) \to n^-_{2\cdot}(T)$
$\mathcal{O}_9:$
$n^{s_1}_{2\cdot}(2T)$
$n^{s_2}_{2\cdot}(2T)$
$\mathcal{O}_{11}: p^{s_1}_2(0) \to p^{s_1}_2(T) \quad p^{s_2}_2(0) \to p^{s_2}_2(T)$
$p^{s_1}_2(2T)$
$p^{s_2}_2(2T)$
$\mathcal{O}_{13}:$
$n^{s_1}_{2\cdot}(3T)$
$n^{s_2}_2(3T)$
$p_2(T)$
$\tilde{n}^\pm(3T)$
$w^\pm_2(T)$
$\mathcal{O}_{15}: \ \tilde{n}^+(0) \to \tilde{n}^-(T) \quad \tilde{n}^-(0) \to \tilde{n}^-(T)$
$p^{s_1}_i(3T), p^{s_2}_i(3T)$
$\mathcal{O}_{17}:$
$\tilde{n}^{s_1}(2T)$
$\tilde{n}^{s_2}(2T)$
$\mathcal{O}_{19}: y^{s_1}(0) \to y^{s_1}(T) \quad y^{s_2}(0) \to y^{s_2}(T)$
$y^{s_1}(3T), y^{s_2}(3T)$
$y^{s_1}(2T)$
$y^{s_2}(2T)$
$y_l(T)$
$\mathcal{O}_{21}:$
$\tilde{n}^{s_1}(3T)$
$\tilde{n}^{s_2}(2T)$
$y(T)$
Feedforward Module
propagate via initial conditions
propagate via parameters

Fig. 3. Schema of the propagation of error via initial concentrations and parameters from the $m-1$th to the mth iteration in the feedforward Module. The top two blocks represent the hidden layer neurons, and the bottom one denotes the output neuron.

1) Error in lwsBCRN: Both types of errors are present in $\mathcal{M}^f_{l_1}$. The current error should be obtained from the proven exponential stability of the lwsBCRN, and as displayed in Fig. 3, the accumulative errors stem from the inaccuracies in parameters $s(T)$, $w^\pm_1(0)$ (concentrations of catalysts) and initial conditions $n^\pm(0)$. Here, only the vertical propagation from the input species needs consideration, and the estimation is presented below.

Lemma 1: Let $n^{+,*} = \mathcal{N}^1_+$, $n^{-,*} = \mathcal{N}^1_-$ denote the positive and negative part of the desired $\mathcal{N}^1$, then there exist $m^+_4, m^-_4 > 0$ that rely on training samples and initial $\mathcal{W}_1$ in FCNN such that the TE_1 for computing $\mathcal{N}^1_\pm$ satisfying $\|n^\pm(T) - n^{\pm,*}\| \leq m^\pm_4 e^{-T}$.

2) Error in sigBCRN: As depicted in Fig. 3, the errors in calculating Υ^1 result from the propagation through the computational reaction subsystems $\mathcal{M}^f_{n^j_1}$ that occur in phases $\mathcal{O}_9$, $\mathcal{O}_{11}$, and $\mathcal{O}_{13}$, separately, and there contain two sigBCRNs corresponding to the two nodes in the hidden layer. The module $\mathcal{M}^f_{n^1_1}$ in phase $\mathcal{O}_9$ contributes to annihilate $\{N^+_{il}\}$ and $\{N^-_{il}\}$ to obtain the difference. According to Lemma 1 in [19], initial concentrations of $\mathcal{M}^f_{n^1_1}$ determine the compatibility class, leading to two types of LSSs. Thus, the accumulative error here is entirely attributed to initial conditions arising from the vertical propagation from $\mathcal{M}^f_{l_1}$. We give a consolidated result to illustrate the error generated in phase $\mathcal{O}_9$.

Lemma 2: Let $|\mathcal{N}^1_{il}| = |\{n^{+,*}_{il}\}_{\mathcal{O}_7} - \{n^{-,*}_{il}\}_{\mathcal{O}_7}|$ denotes the absolute value of desired $\mathcal{N}^1_{il}$. We can find $m_4 = m^+_4 + m^-_4$,

$m_j^{'i}, \alpha_j^{'i} > 0$ related to $\{n_{il}^{\pm,*}\}_{\mathcal{O}_7}$ with $i, j \in \{1,2\}$ such that

$$|n_{il}^{s_1}(2T) - |\mathcal{N}_{il}^1|| \leq m_4 e^{-T} + m_j^{'i} e^{-\alpha_j^{'i}}$$
$$|n_{il}^{s_2}(2T) - 0| \leq m_j^{'i} e^{-\alpha_j^{'i}}$$

where if $\mathcal{N}_{il}^1 > 0$, we have $j = 1, s_1 = +, s_2 = -$, otherwise $j = 2, s_1 = -, s_2 = +$.

In effect, the estimation above is subject to a hidden assumption. It remains valid when the propagation errors on initial conditions from $\mathcal{O}_7$ are unable to alter the targeted attractive region of the annihilation reaction in $\mathcal{M}_{n_1^1}^f$. However, intuitively, another case cannot be dismissed if the error significantly impacts the attractive region. Thus, we present the following lemma to show that this scenario relies on T, and an appropriate phase length is feasible to identify to mitigate the worst case.

Lemma 3: Consider these two alternating operating systems

$$\Sigma_1\!:\!\begin{array}{l}\dot{x}(t) = c_1^m - x(t)\\ \dot{y}(t) = c_2^m - y(t),\end{array} \Sigma_2\!:\!\begin{array}{l}\dot{x}(t) = -x(t)y(t)\\ \dot{y}(t) = -x(t)y(t)\end{array}$$

where constants c_1^m, c_2^m will change over the round m. Systems $\sum_1, \sum_2$ can separately last for any fixed time length T in every round. Given the initial condition $x_0^1 = y_0^1 = 0$ in the first round, if $T > \ln(1 + \frac{\sum_{j=1}^{m-1}|c_1^{m-j} - c_2^{m-j}|}{|c_1^m - c_2^m|})$, it holds that $(c_1^m - c_2^m)(x(2mT) - y(2mT)) \geq 0$ for all m.

Remark 2: It is possible to provide a rough range estimation for T-based solely on the information of FCNN since $c_1^m = (w_1^+(0)s_a(T))_{il}$, $c_2^m = (w_1^-(0)s_a(T))_{il}$ in BFCNN are close to the desired values $\{\mathcal{N}_{+,il}^1\}_{\mathcal{O}_7}, \{\mathcal{N}_{-,il}^1\}_{\mathcal{O}_7}$. The right-hand side of (S3) requires T to surpass 15–20 while considering the weights range and typical iteration numbers in computers, which is not a tough requirement for the oscillator.

We now address the errors in $\mathcal{M}_{n_1^2}^f$ with equations for i, l

$$\mathcal{M}_{n_1^2}^f : \dot{p}_{il}^{\pm}(t) = n_{il}^{\pm}(t)\left(\frac{1}{2} - p_{il}^{\pm}(t)\right)$$

under the condition that T is designed within the valid range above. The discussion is twofold. In the case of $\mathcal{N}_{il}^1 > 0$, we present the distance between the trajectory and the desired value $\frac{1}{2}$ as $|p_{il}^+(t) - \frac{1}{2}| = |p_{il}^+(0) - \frac{1}{2}|e^{-n_{il}^+(2T)t}$, from which the current error and propagation error stored in $n_{il}^+(2T)$ (as the constant parameter here) hinders the hidden-layer output species P_{il}^+ from achieving the target value at the original rate. In other words, errors emerging in the power item impact the achievable accuracy of systems within a fixed time. Similar arguments can be applied in the negative case. As for $\mathcal{M}_{n_1^3}^f$, recall that the dynamical equations is designed as [19]

$$\mathcal{M}_{n_1^3}^f : \dot{p}_{il}^+(t) = n_{il}^+(t)p_{il}^+(t)(1 - p_{il}^+(t)),\ \dot{n}_{il}^+(t) = -n_{il}^+(t)$$
$$\dot{p}_{il}^-(t) = n_{il}^-(t)p_{il}^-(t)(p_{il}^-(t) - 1),\ \dot{n}_{il}^-(t) = -n_{il}^-(t)$$
$$\dot{p}_{il}(t) = p_{il}^+(t) + p_{il}^-(t) - p_{il}(t).$$

In this module, the accumulative errors, strongly depending on $p_{il}^{\pm}(T)$ and $n_{il}^{\pm}(2T)$, influence both the original convergence characteristic (the current error needs to be assigned coefficients with new valid ranges) and the LSSs. Therefore, incorporating Lemma 2, we give the total error bound stemming from computing hidden-layer outputs Υ^1 below.

Proposition 2: Let $p_{il}^* = \Upsilon_{il}^1 > 0$ be the desired value of hidden-layer outputs, and s_1, s_2 be sign markers. There exist $m_{5i}^{\pm}, m_{6i}^{\pm}, \tilde{m}_{il}^{\pm}, a_{qi}^{\pm} > 0$ with $i \in \{1,2\}, q = \{4,5,6\}$ such that

$$|p_{il}^{s_1}(2T) - \Upsilon_{il}^1| < \left(\tilde{m}_{il}^{s_1} + \frac{m_4}{4}\right)e^{-T} + \frac{m_j^{'i}}{4}e^{-\alpha_j^{'i}T}$$
$$+ m_{6i}^{s_1}m_{5i}^{s_1}(m_j^{'i}Te^{-\alpha_j^{'i}T} + m_4Te^{-T} + e^{-|\mathcal{N}_{il}^1|T})$$
$$|p_{il}^{s_2}(2T) - 0| \leq \frac{m_j^{'i}}{4}e^{-\alpha_j^{'i}T} + m_{5i}^{s_2}m_j^{'i}Te^{-\alpha_j^{'i}T}$$
$$|p_{il}(T) - p_{il}^*| < a_{4i}^{s_1}Te^{-T} + a_{5i}^{s_1}Te^{-\alpha_j^{'i}T} + a_{6i}^{s_1}e^{-|\mathcal{N}_{il}^1|T}$$

where $j \in \{1,2\}$, and if $\mathcal{N}_{il}^1 > 0$, $s_1 = +, s_2 = -, j = 1$, otherwise $s_1 = -, s_2 = +, j = 2$.

Even though expanding items in parameters $\{a_{4i}^{\pm}, a_{5i}^{\pm}, a_{6i}^{\pm}\}$ here is complicated, from the derivation, we can conclude that they are related to training samples and the choice of initial weights $\mathcal{W}_1$ in FCNN.

3) *Error in the Output Layer:*

We have clarified the error propagation pattern within lwsBCRN and sigBCRN, as well as estimated the errors in the linear input and nonlinear output of the hidden layer nodes. As displayed in Fig. 3, the vertical propagation is identical for each layer in BFCNN due to the uniform connectivity structure across all feedforward layers. Therefore, in this part, we derive the TE_1 bound of $|y_l(T) - y_l^*|$ following the same workflow. This estimation provides an upper bound on the deviation between the final concentration of the output species $\{Y^l\}$ and the ideal value y_l after a single feedforward computation in the BFCNN. We utilize $\tilde{n}^{\pm}(t) \in \mathbb{R}_{\geq 0}^{1\times\tilde{p}}$, $y(t)$, $y^{\pm}(t) \in \mathbb{R}_{\geq 0}^{1\times\tilde{p}}$, to denote concentration vectors of the net input species $\{N_{3l}^{\pm}\}$, output species $\{Y^l\}$, and output species pairs $\{Y_l^{\pm}\}$, respectively, introduce the matrix variable $p_a(t) = ((p(t))^{\top}, \mathbb{1})^{\top}$ for the subsequent convenience, and display the result for the output layer without illustration.

Corollary 1: Suppose that $\tilde{n}^{+,*} = \tilde{\mathcal{N}}_+^1$ and $\tilde{n}^{-,*} = \tilde{\mathcal{N}}_-^1$ represent the positive and negative part of desired $\tilde{\mathcal{N}}^1$, and s pertains to the sign marker. Then, for $\forall l \in \{1,\ldots,\tilde{p}\}$, there exist $m_7^{\pm}, a_q, \tilde{e}_{\tilde{q}} > 0$ with $q \in \{4,5,6\}$ and $\tilde{q} \in \{1,\ldots,5\}$ and powers $\alpha, \alpha_3, n^1 > 0$ for the following estimation:

$$\|\tilde{n}^{\pm}(T) - \tilde{n}^{\pm,*}\| \leq m_7^{\pm}(a_4Te^{-T} + a_5Te^{-\alpha T} + a_6e^{-n^1T})$$
$$|y_l(T) - y_l^*| < \tilde{e}_1Te^{-\alpha_3T} + \tilde{e}_2T^2e^{-T} + \tilde{e}_3T^2e^{-\alpha T}$$
$$+ \tilde{e}_4Te^{-n^1T} + \tilde{e}_5e^{-|\tilde{\mathcal{N}}_l^1|T}$$

where $\{a_5, \alpha\} = \arg\max_{\pm,j,i}\{2a_{5i}^{\pm}Te^{-\alpha_j^{'i}T}\}$, $m_7^{\pm} = 2\|w_2^{\pm}(0)\|$, $a_4 = \max_{\pm,i}\{2a_{4i}^{\pm}\} + \max_{\pm}\{\frac{\|w_2^{\pm}(0)\{p_a^*\}_{\mathcal{O}_{13}} - \tilde{n}^{\pm}(0)\|}{m_7^{\pm}}\}$, and $\{a_6, n^1\} = \arg\max_{\pm,i,l}\{2a_{6i}^{\pm}Te^{-|\mathcal{N}_{il}^1|T}\}$.

Similarly, α_3 depends on $\{\tilde{n}_l^{\pm,*}\}_{\mathcal{O}_{15}}$ and $\tilde{e}_{\tilde{q}}$ are related to hidden-layer outputs and the choice of initial $\mathcal{W}_2$ in FCNN. Corollary 1 provides the cumulative result of vertical error propagation in a single feedforward computation. This upper bound for the output layer serves as a critical input error in the estimation for the learning part, as the prediction output of BFCNN is key to the backpropagation training process. Hereafter, we write $\{A\} = a_4 T e^{-T} + a_5 T e^{-\alpha T} + a_6 e^{-n^1 T}$ to denote the error bound for computing Υ, $\{E\}$ to indicate the bound for computing y_l^* when $m = 1$, and $\{A\}^m$, $\{E\}^m$ to represent the bounds of the same form for the mth iteration.

C. Preceding Calculation Module Error

The pBCRN, solely occupying $\mathcal{O}_{23}$, comprises boundary equilibrium reactions $\mathcal{O}_{23}^1$ and type-I CRNs $\mathcal{O}_{23}^2$, $\mathcal{O}_{23}^3$ [19]. Recall that for $i = \{1, 2\}$, $y_e(t), y_s(t), \tilde{y}(t), i_y(t) \in \mathbb{R}_{\geq 0}^{1\times\tilde{p}}$ and $p_{is}(t), \tilde{p}_i(t), i_{p_i}(t) \in \mathbb{R}_{\geq 0}^{1\times\tilde{p}}$ indicates concentration vectors of $\{Y_e^l, Y_s^l, \tilde{Y}^l, I_y^l\}$ of MAS $\mathscr{M}^1$, and $\{P_{is}^l, \tilde{P}_i^l, I_{p_i}^l\}$ of $\mathscr{M}_{(i)}^2$, respectively, in $\mathcal{O}_{23}^1$. The dynamical equations with rate constant k are

$$\begin{aligned}
\mathscr{M}^1 :\ & \dot{y}_l(t) = -k y_l(t),\ \dot{\tilde{y}}_l(t) = k y_l(t)\\
& \dot{y}_e^l(t) = k y_l(t) - k y_e^l(t) s_3^l(t),\ \dot{s}_3^l(t) = -k y_e^l(t) s_3^l(t)\\
& \dot{y}_s^l(t) = k y_l(t) - k y_s^l(t) i_y^l(t),\ \dot{i}_y^l(t) = -k y_s^l(t) i_y^l(t)\\
\mathscr{M}_{(i)}^2 :\ & \dot{p}_{il}(t) = -k p_{il}(t),\ \dot{p}_{is}^l(t) = k p_{il}(t) - k p_{is}^l(t) i_{p_i}^l(t)\\
& \dot{\tilde{p}}_{il}(t) = k p_{il}(t),\ \dot{i}_{p_i}^l(t) = -k p_{is}^l(t) i_{p_i}^l(t).
\end{aligned}$$

Then, $e^{\pm}(t)$, $s_y^l(t)$, $s_{p_i}^l(t)$, $e(t) \in \mathbb{R}^{1\times\tilde{p}}$ separately denote the concentration vectors of species $\{E_l^{\pm}\}$, $\{S_y^l\}$, $\{S_{p_i}^l\}$, $\{E_l\}$ in $\mathcal{O}_{23}^2$, $\mathcal{O}_{23}^3$ with dynamical equations according to type-I CRN

$$\begin{aligned}
\mathcal{O}_{23}^2 :\ & \dot{e}_l^+(t) = s_3^l(t) - e_l^+(t),\ \dot{e}_l^-(t) = y_e^l(t) - e_l^-(t)\\
& \dot{s}_y^l(t) = i_y^l(t) - s_y^l(t),\ \dot{s}_{p_i}^l(t) = i_{p_i}^l(t) - s_{p_i}^l(t)\\
\mathcal{O}_{23}^3 :\ & \dot{e}_l(t) = e_l^+(t) + e_l^-(t) - e_l(t).
\end{aligned}$$

Part I elaborates that there exist $M_x, M_y > 0$, relying on initial concentrations, and exponential convergence orders $\tilde{K}, K^i > 0$, relying on initial concentrations and k, for fixed T of $\mathscr{M}^1$, $\mathscr{M}_{(i)}^2$ with boundary LSSs, which facilitate to estimate the current error of $\mathcal{O}_{23}^1$. For $\mathcal{O}_{23}^2$ and $\mathcal{O}_{23}^3$, following the derivation in Lemma 2 of Part I, the concrete current errors are combined into two scenarios, i.e., if $\tilde{K}, K^i \neq 1$, we have

$$\begin{aligned}
\|x_e^l(T) - \bar{x}_e^l\| &< n_x(e^{-\tilde{K}T} - e^{-T}) + g_x e^{-T}\\
|s_{p_i}^l(T) - \bar{s}_{p_i}^l| &< n_y^i(e^{-K^i T} - e^{-T}) + g_{p_i} e^{-T}\\
|e_l(T) - \bar{e}_l| < 2\tilde{n}_x e^{-\tilde{K}T} &+ (g^+ + g^- - 2n_x) T e^{-T} + g_e e^{-T}
\end{aligned} \tag{11}$$

where $x_e^l(t) = (e_l^+(t), e_l^-(t), s_y^l(t))^\top$, $n_x = \frac{M_x}{1-\tilde{K}}$, $n_y^i = \frac{M_y^i}{1-K^i}$, $\tilde{n}_x = \frac{M_x}{(1-\tilde{K})^2}$, $g^{\pm} = |e_l^{\pm}(0) - \bar{e}_l^{\pm}|$, $g_{p_i} = |s_{pi}^l(0) - \bar{s}_{pi}^l|$, $g_e = |e_l(0) - \bar{e}_l|$, and $g_x = \max\{g^{\pm}, |s_y^l(0) - \bar{s}_y^l|\}$ and if $\tilde{K}, K^i = 1$, they become

$$\begin{aligned}
\|x_e^l(T) - \bar{x}_e^l\| &< M_x T e^{-T} + g_x e^{-T}\\
|s_{p_i}^l(T) - \bar{s}_{p_i}^l| &< M_y^i T e^{-T} + g_{p_i} e^{-T}
\end{aligned}$$

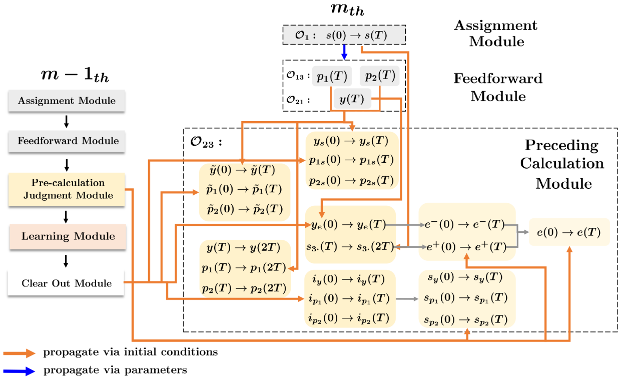
m_{th}
$\mathcal{O}_1:\ s(0) \to s(T)$
Assignment Module
$\mathcal{O}_{13}:\ p_1(T)\quad p_2(T)$
$\mathcal{O}_{21}:\ y(T)$
Feedforward Module
$m-1_{th}$
Assignment Module
Feedforward Module
Pre-calculation Judgment Module
Learning Module
Clear Out Module
$\mathcal{O}_{23}:$
$y_s(0) \to y_s(T)$
$p_{1s}(0) \to p_{1s}(T)$
$p_{2s}(0) \to p_{2s}(T)$
Preceding Calculation Module
$\tilde{y}(0) \to \tilde{y}(T)$
$\tilde{p}_1(0) \to \tilde{p}_1(T)$
$\tilde{p}_2(0) \to \tilde{p}_2(T)$
$y_e(0) \to y_e(T)$
$s_{3\cdot}(T) \to s_{3\cdot}(2T)$
$e^-(0) \to e^-(T)$
$e^+(0) \to e^+(T)$
$e(0) \to e(T)$
$y(T) \to y(2T)$
$p_1(T) \to p_1(2T)$
$p_2(T) \to p_2(2T)$
$i_y(0) \to i_y(T)$
$i_{p_1}(0) \to i_{p_1}(T)$
$i_{p_2}(0) \to i_{p_2}(T)$
$s_y(0) \to s_y(T)$
$s_{p_1}(0) \to s_{p_1}(T)$
$s_{p_2}(0) \to s_{p_2}(T)$
propagate via initial conditions
propagate via parameters

Fig. 4. Schema of the propagation of error via initial concentrations and parameters from the $m-1$th to the mth iteration in the preceding calculation module.

$$|e_l(T) - \bar{e}_l| < M_x T^2 e^{-T} + (g^+ + g^-) T e^{-T} + g_e e^{-T}. \tag{12}$$

Now, we discuss the total error bound of every type of species. The error propagation schematic is shown in Fig. 4. First, since no errors accumulated leading to the deviation of the LSSs of species $\{Y^l, P_i^l\}$ that participate in pBCRN, based on the available analytic solutions, we have

$$\begin{aligned}
|y_l(2T) - 0| &= |y_l(T)| e^{-kT} < y_l^* e^{-kT} + \{E\} e^{-kT}\\
|p_{il}(2T) - 0| &= |p_{il}(T)| e^{-kT} < \Upsilon_{il}^1 e^{-kT} + \{A\} e^{-kT}.
\end{aligned}$$

Then, in our design, the equilibrium concentrations of species $\{\tilde{Y}^l, \tilde{P}^l\}$ are used to replace that of the original output species $\{Y^l, P_i^l\}$ for further update computation. Thus, their deviation from the desired value is given closer attention below.

Lemma 4: Suppose that the rate constant is taken as $k > \max\{1, \tilde{n}^1, \alpha, |\tilde{\mathcal{N}}_l^1|, \alpha_3\}$, the errors in obtaining y_l^*, Υ_{il}^1 caused by $\{\tilde{Y}^l, \tilde{P}_i^l\}$ are of the same order as errors resulting from species $\{Y^l, P_i^l\}$.

As for other species in $\mathcal{O}_{23}^1$, the estimates are as follows.

Lemma 5: For $l \in \{1, \ldots, \tilde{p}\}$ and $i = \{1, 2\}$, let $s_3^{l,*}, y_e^{l,*}$ denote desired values computed by species $\{S_3^l, Y_e^l\}$ in $\mathcal{O}_{23}^1$, then the error upper bounds of species $\{S_3^l, Y_e^l\}$ and $\{Y_s^l, I_y^l\}$, $\{P_{is}^l, I_{p_i}^l\}$ become

$$\begin{aligned}
|y_s^l(T) - 0| < M_x e^{-\tilde{K}T},\ & |p_{is}^l(T) - 0| < M_y^i e^{-K^i T}\\
|i_y^l(T) - (1 - y_l^*)| &< M_x e^{-\tilde{K}T} + \{E\}\\
|i_{p_i}^l(T) - (1 - \Upsilon_{il}^1)| &< M_y^i e^{-K^i T} + \{A\}\\
|y_e^l(T) - y_e^{l,*}| < M_x e^{-\tilde{K}T} &+ \mathbb{1}_{\{(y_l(T) > s_3^l(T))\}}(m_1 e^{-T} + \{E\})\\
|s_3^l(2T) - s_3^{l,*}| < M_x e^{-\tilde{K}T} &+ \mathbb{1}_{\{(y_l(T) < s_3^l(T))\}}(m_1 e^{-T} + \{E\})
\end{aligned}$$

where $\mathbb{1}$ denotes the indicator function.

Remark 3: Mild assumptions should be imposed on the fixed T to avoid the radius of deviation of actual concentrations from desired values leading to $y_e^l(0) + y_l^1(T) > s_3^l(T)$ while $y_l^* < d^{l,*}$, or $y_e^l(0) + y_l^1(T) < s_3^l(T)$ while $y_l^* > d^{l,*}$, and causing $y_l(T) + y_s^l(0)$, $p_{il}(T) + p_{is}^l(0) > 1$, which implies that errors alter the originally intended attractive regions. Here, T is assumed to be taken in the implicit feasible range

$$\{E\} + m_1 e^{-T} < |d^{l,*} - y_l^*|, \ \{E\} < 1 - y_l^*$$

if $m = 1$. The range suggests that T needs to satisfy another lower bound, similar to the one presented in Lemma 3.

Regarding the type-I CRN in $\mathcal{O}_{23}^2$, errors caused by $\{S_y^l, S_{p_i}^l\}$ depends on $K^1, K^2, \tilde{K}$, i.e., the upper bound is

$$\begin{aligned}|s_y^l(T) - (1 - y_l^*)| &< |s_y^l(T) - \bar{s}_y^l| + |\bar{i}_y - i_y^{l,*}| \\ &< \mathbb{I}_{\{\tilde{K}\neq 1\}} n_x (e^{-\tilde{K}T} - e^{-T}) + g_y e^{-T} \\ &\quad + \mathbb{I}_{\{\tilde{K}=1\}} M_x T e^{-T} + \{E\} \\ |s_{p_i}^l(T) - (1 - \Upsilon_{il}^1)| &< |s_{p_i}^l(T) - \bar{s}_{p_i}^l| + |\bar{i}_{p_i}^l - i_{p_i}^{l,*}| \\ &< \mathbb{I}_{\{K^i\neq 1\}} n_y^i (e^{-K^i T} - e^{-T}) + g_{p_i} e^{-T} \\ &\quad + \mathbb{I}_{\{K^i=1\}} M_y^i T e^{-T} + \{A\}.\end{aligned}$$

Note that $\bar{s}_y^l = \bar{i}_y^l, \bar{s}_{p_i}^l = \bar{i}_{p_i}^l, g_y = |s_y^l(0) - i_y^{l,*}| + \{E\}$. As for error species pairs $\{E_l^+, E_l^-\}$, the upper bound should be estimated in four cases, as there are two different scenarios for both current and accumulative errors with $\bar{e}_l^+ = \bar{s}_3^l, \bar{e}_l^- = \bar{y}_e^l$. To elaborate, combining (11) and (12), we derive

$$\begin{aligned}|e_l^+(T) - e_l^{+,*}| &< \mathbb{I}_{\{\tilde{K}=1\}} M_x T e^{-T} + \mathbb{I}_{\{\tilde{K}\neq 1\}} n_x (e^{-\tilde{K}T} - e^{-T}) \\ &\quad + g^+ e^{-T} + \mathbb{I}_{\{y_l^* < d^{l,*}\}} (m_1 e^{-T} + \{E\}) \\ |e_l^-(T) - e_l^{-,*}| &< \mathbb{I}_{\{\tilde{K}=1\}} M_x T e^{-T} + \mathbb{I}_{\{\tilde{K}\neq 1\}} n_x (e^{-\tilde{K}T} - e^{-T}) \\ &\quad + g^- e^{-T} + \mathbb{I}_{\{y_l^* > d^{l,*}\}} (m_1 e^{-T} + \{E\}).\end{aligned}$$

Now, considering $\mathcal{O}_{23}^3$, desired absolute values of errors $e_l^* = |e_l^1| = |d^l - y_l^*|$ for all l are finally stored in species $\{E_l\}$ with $\bar{e}_l = \bar{e}_l^+ + \bar{e}_l^-$ and the realization error is described as

$$|e_l(T) - e_l^*| < |e_l(T) - \bar{e}_l| + m_1 e^{-T} + \{E\}.$$

The upper bound holds whatever the sign of errors and may differ in current error relying on $\tilde{K}$, as (11) and (12). Notably, similar to g_y, propagation errors from $s_3^l(T), p_{il}(T), y_l(T)$ stored on LSSs make coefficients become $g^+ = |e_l^+(0) - s_3^{l,*}| + \mathbb{I}_{\{e_l^1 > 0\}} \epsilon_e$, $g^- = |e_l^-(0) - y_e^{l,*}| + \mathbb{I}_{\{e_l^1 < 0\}} \epsilon_e$, $g_{p_i} = |s_{pi}^l(0) - (1 - \Upsilon_{il}^1))| + \{A\}$, and $g_e = |e_l(0) - e_l^*| + \epsilon_e$ with $\epsilon_e = m_1 e^{-T} + \{E\}$.

Realization errors stemming from the Judgment Module are not treated in isolation within the scope of this article. The total errors impact the reaction rates in the learning module without explicitly altering their LSSs based on the design [19]. This influence is akin to that of a chemical oscillator. It can be mitigated by adjusting subsequent reaction rate constants and adding clear-out reactions to zero some concentrations manually.

D. Learning Module Error

The learning module comprises the computational reaction subsystems $\mathcal{M}_1^{le}$ designed for computing negative gradients and $\mathcal{M}_2^{le}$ intended for obtaining updated weights, operating during phases $\mathcal{O}_{27}$ and $\mathcal{O}_{29}$, separately. The current error and accumulative error in $\mathcal{M}_1^{le}$ are both segmented into the multiplication component (arising from $\mathcal{O}_{27}^1$) and the addition component (arising from $\mathcal{O}_{27}^2$). Notably, as illustrated in Fig. 5, almost every factor in (5) computed by the upstream BCRN modules is inaccurate, resulting in the accumulative error on LSSs of $\mathcal{O}_{27}^1$ through equation parameters. Now we present a general upper bound estimation for such multiplication errors.

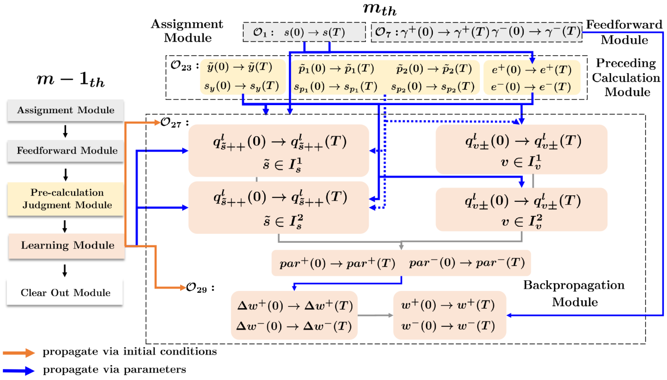
m_{th}
Assignment Module
$\mathcal{O}_1: \ s(0) \to s(T)$
$\mathcal{O}_7: \gamma^+(0) \to \gamma^+(T) \gamma^-(0) \to \gamma^-(T)$
Feedforward Module
$\mathcal{O}_{23}:$
$\tilde{y}(0) \to \tilde{y}(T)$
$s_y(0) \to s_y(T)$
$\tilde{p}_1(0) \to \tilde{p}_1(T)$
$s_{p_1}(0) \to s_{p_1}(T)$
$\tilde{p}_2(0) \to \tilde{p}_2(T)$
$s_{p_2}(0) \to s_{p_2}(T)$
$e^+(0) \to e^+(T)$
$e^-(0) \to e^-(T)$
Preceding Calculation Module
$m - 1_{th}$
Assignment Module
Feedforward Module
Pre-calculation Judgment Module
Learning Module
Clear Out Module
$\mathcal{O}_{27}:$
$q_{\tilde{s}++}^l(0) \to q_{\tilde{s}++}^l(T)$
$\tilde{s} \in I_s^1$
$q_{\tilde{s}++}^l(0) \to q_{\tilde{s}++}^l(T)$
$\tilde{s} \in I_s^2$
$q_{v\pm}^l(0) \to q_{v\pm}^l(T)$
$v \in I_v^1$
$q_{v\pm}^l(0) \to q_{v\pm}^l(T)$
$v \in I_v^2$
$par^+(0) \to par^+(T)$
$par^-(0) \to par^-(T)$
$\mathcal{O}_{29}:$
$\Delta w^+(0) \to \Delta w^+(T)$
$\Delta w^-(0) \to \Delta w^-(T)$
$w^+(0) \to w^+(T)$
$w^-(0) \to w^-(T)$
Backpropagation Module
propagate via initial conditions
propagate via parameters

Fig. 5. Schema of the propagation of error via initial concentrations and parameters from the $m - 1$th to the mth iteration in the Learning Module.

Lemma 6: Suppose that we have $|\bar{x}_i - x_i^*| < \epsilon_i$ for $\forall i \in \{1, 2, \ldots, n\}$, where x_i^* is constant, $\epsilon_i > 0$ denotes a sufficiently small upper bound, and $\bar{x} = [\bar{x}_1, \ldots, \bar{x}_n]^\top \in \Omega$, a bounded set in $\mathbb{R}^n$. Then, there exists a set of parameters $\{h_i\}_{i=1}^n > 0$ such that $|\prod_{i=1}^n \bar{x}_i - \prod_{i=1}^n x_i^*| < \sum_{i=1}^n h_i \epsilon_i$.

Lemma 6 concludes that the multiplication error is the sum of initial errors of all factors, contributing to derive the accumulative error of monomial species $\{Q_{s\pm\pm}^l, Q_{v\pm}^l\}$ for $s \in I_s = I_s^1 \bigcup I_s^2 = \{1, 2, 3, 4\} \bigcup \{7, 8\}$ and $v \in I_v = I_v^1 \bigcup I_v^2 = \{5, 6\} \bigcup \{9\}$ with concentrations $q_{s\pm\pm}^l(t), q_{v\pm}^l(t)$ in $\mathcal{O}_{27}^1$. Taking Q_{1++}^l as an example, we have the dynamical equations for computing $q_{1++}^l = e_l^+ \mathcal{W}_{2_{11}}^+ y_l (1 - y_l) \tilde{\Upsilon}_{1\,l} (1 - \tilde{\Upsilon}_{1\,l}) \Xi_{1\,l}$ that

$$\begin{aligned}&\dot{u}_1(t) = e_l^+(t) p_{1l}^+(t) - u_1(t), \ \dot{u}_2(t) = \tilde{y}^l(t) s_y^l(t) - u_2(t) \\ &\dot{u}_3(t) = s_{p_i}^l(t) w_{2_{11}}^+(t) - u_3(t), \ \dot{\tilde{u}}_1(t) = u_1(t) u_2(t) - \tilde{u}_1(t) \\ &\dot{\tilde{u}}_2(t) = s_1^l(t) u_3(t) - \tilde{u}_2(t), \ \dot{q}_{1++}^l(t) = \tilde{u}_1(t) \tilde{u}_2(t) - q_{1++}^l(t)\end{aligned}$$

where $u_i(t), \tilde{u}_j(t)$ denotes concentrations of intermediate species in $\mathcal{O}_{27}^1$ and we direct readers to Part I for more details. This approach is effective since initial errors of factors constituting $\bar{q}_{1++}^l = e_l^+(T) \tilde{p}_{1\,l}(T) \tilde{y}_l(T) s_y^l(T) s_{p_i}^l(T) w_{2_{11}}^+(0) s_1^l(T)$ have been previously derived. Here, we introduce $par^\pm(t) \in \mathbb{R}_{\geq 0}^{3\times 3}$ to denote the concentration matrix of negative species pairs $\{\mathrm{Par}_{\tilde{i}}^\pm\}_{\tilde{i}=1}^9$ in $\mathcal{O}_{27}^2$ with the correspondence between $\tilde{i}$ and the (i, j) element in $\mathrm{par}^\pm(t)$ following $1, 3, 7 \leftrightarrow 11, 12, 13$, $2, 4, 8 \leftrightarrow 21, 22, 23$, $5, 6, 9 \leftrightarrow 31, 32, 33$, respectively. We illustrate dynamics of $\mathcal{O}_{27}^2$ by species $\{\mathrm{Par}_1^\pm\}$

$$\dot{\mathrm{par}}_{11}^\pm(t) = \sum_{l=1}^{\tilde{p}} (q_{1+\pm}^l(t) + q_{1-\mp}^l(t)) - \mathrm{par}_{11}^\pm(t).$$

Then, by combining the exponential convergence characterization of type-II CRN (see [19, Lemma 3]) and incorporating the addition component, we present total errors of $-\nabla\mathcal{E}_{+}^{1}$ and $-\nabla\mathcal{E}_{-}^{1}$ parts of the desired negative gradient of the training error $\mathcal{E}^1$, computed in $\mathcal{M}_1^{le}$, for all weights.

Proposition 3: Let $((\mathrm{par}_{1,\pm}^{*})^{\top}, (\mathrm{par}_{2,\pm}^{*})^{\top}))^{\top} = \mathrm{par}_{\pm}^{*} = -\nabla\mathcal{E}_{\pm}^{1}$ with $\mathrm{par}_{1,\pm}^{*} \in \mathbb{R}_{\geq 0}^{2\times 3}$, $\mathrm{par}_{2,\pm}^{*} \in \mathbb{R}_{\geq 0}^{1\times 3}$ be the block matrices of $-\nabla\mathcal{E}_{\pm}^{1}$. Then, there exist the positive parameter set $\{\mathcal{L}_s^q\}_{q=1}^{6}$, $\{\mathcal{L}_{s,i}^{7}\}_{i=1}^{2}$, $\{\mathcal{L}_{\pm,s}^{8}\}$, and $\tilde{\gamma} = 1 - \frac{1}{e}$ for species $\{\mathrm{Par}_s^{\pm}\}$ in $\mathcal{O}_{27}^{2}$ with $s \in I_s$ such that

$$\begin{aligned}|\mathrm{par}_{1_{ij}}^{\pm}(T) - \mathrm{par}_{1,\pm_{ij}}^{*}| &< \mathcal{L}_s^1 e^{-\tilde{K}T} + \mathcal{L}_s^2 T e^{-\alpha_3 T}\\ &+ \mathcal{L}_s^3 T^2 e^{-T} + \mathcal{L}_s^4 T^2 e^{-\alpha T} + \mathcal{L}_s^5 T e^{-n^1 T}\\ &+ \mathcal{L}_s^6 e^{-\tilde{n}^1 T} + \mathcal{L}_{s,i}^{7} e^{-K^i T} + \mathcal{L}_{\pm,s}^{8} e^{-\tilde{\gamma}T}. \end{aligned} \tag{13}$$

Concerning species $\{\mathrm{Par}_v^{\pm}\}$ with $v \in I_v$, we can find another positive parameter set $\{\tilde{\mathcal{L}}_{\pm,v}^{q}\}_{q=1}^{7}$ for the error estimation

$$\begin{aligned}|\mathrm{par}_{2_{1j}}^{\pm}(T) - \mathrm{par}_{2,\pm_{1j}}^{*}| &< \tilde{\mathcal{L}}_{\pm,v}^{1} e^{-\tilde{K}T} + \tilde{\mathcal{L}}_{\pm,v}^{2} T e^{-\alpha_3 T}\\ &+ \tilde{\mathcal{L}}_{\pm,v}^{3} T^2 e^{-T} + \tilde{\mathcal{L}}_{\pm,v}^{4} T^2 e^{-\alpha T} + \tilde{\mathcal{L}}_{\pm,v}^{5} T e^{-n^1 T}\\ &+ \tilde{\mathcal{L}}_{\pm,v}^{6} e^{-\tilde{n}^1 T} + \tilde{\mathcal{L}}_{\pm,v}^{7} e^{-\tilde{\gamma}T} \end{aligned} \tag{14}$$

with $\mathrm{par}^{\pm}(t) = ((\mathrm{par}_1^{\pm}(t))^{\top}, (\mathrm{par}_2^{\pm}(t))^{\top})^{\top}$.

Estimating the distance between the real iterative sequence $w^{\pm}(T) = ((w_1^{\pm}(T))^{\top}, (w_2^{\pm}(T))^{\top})^{\top} \in \mathbb{R}_{\geq 0}^{3\times 3}$ and desired sequence $\mathcal{W}_{\pm}^{2}$ highlights the difference in the training performance of two neural networks. As displayed in Fig. 5, the deviation in computing negative gradients by $\mathcal{M}_1^{le}$ and storing current weight/bias concentrations in species $\{\Gamma_i^{\pm}\}_{i=1}^{9}$ by $\mathcal{O}_7^2$ [19] leads to the accumulative error on incremental weight/bias species $\{\Delta W_i^{\pm}\}_{i=1}^{6}, \{\Delta B_j^{\pm}\}_{j=1}^{3}$ in $\mathcal{M}_2^{le}$ and weight/bias species. Here, $\Delta w^{\pm}(t) \in \mathbb{R}_{\geq 0}^{3\times 3}$ is regarded as the concentration matrix of incremental species with $\Delta w_{1\cdot}^{\pm}(t)$ to identify $\Delta W_1^{\pm}, \Delta W_3^{\pm}, \Delta B_1^{\pm}$'s, $\Delta w_{2\cdot}^{\pm}(t)$ to denote $\Delta W_2^{\pm}, \Delta W_4^{\pm}, \Delta B_2^{\pm}$'s, and $\Delta w_{3\cdot}^{\pm}(t)$ to indicate $\Delta W_5^{\pm}, \Delta W_6^{\pm},$ and $\Delta B_3^{\pm}$'s. The initial deviation of $\Gamma_i^{\pm}$ is

$$\begin{aligned}\|\gamma^{\pm}(T) - \mathcal{W}_{\pm}^{1}\| &< \|\gamma^{\pm}(T) - w^{\pm}(0)\| + \|w^{\pm}(0) - \mathcal{W}_{\pm}^{1}\|\\ &< \|\gamma^{\pm}(T) - w^{\pm}(0)\| < \mathcal{I}^{\pm} e^{-T}\end{aligned}$$

where $\gamma^{\pm}(t) \in \mathbb{R}_{\geq 0}^{3\times 3}$ represent concentration matrices of $\Gamma_i^{\pm}$ and $\mathcal{I}^{\pm} = \|w^{\pm}(0) - \gamma^{\pm}(0)\|$. By combining the established exponential convergence of $\mathcal{M}_2^{le}$ [19], we present the following theorem to conclude the overall deviation in updating weights during the first iteration.

Theorem 4: For the BFCNN given in Table I, assume the initial concentrations to follow that $s(0)$ is an arbitrary positive matrix, $c(0) = (\xi_1^{\top}, \ldots, \xi_{\tilde{p}}^{\top})$, $(x_1^i(0), x_2^i(0), d^i(0))^{\top} = \chi_{.i}$, $\tilde{c}(0) = \mathbb{O}$ in the assignment module; $w^{\pm}(0) = \mathcal{W}_{\pm}^{1}$, $half(0) = 0.5$ in the feedforward module; the initial concentrations of species $\{I_y^l, I_{p_1}^l, I_{p_2}^l\}$ are 1in the pBCRN of the judgment module; in addition, the other species in BFCNN is set to have the initial concentration 0. Then, at $m = 1$, for $w_{\pm,1}^{*,1} = \mathcal{W}_{\pm,1}^{2}$, $w_{\pm,2}^{*,1} = \mathcal{W}_{\pm,2}^{2}$ with $((\mathcal{W}_{\pm,1}^{2})^{\top}, (\mathcal{W}_{\pm,2}^{2})^{\top})^{\top} = \mathcal{W}_{\pm}^{2}$ and $\mathcal{W}_{\pm,1}^{2} \in \mathbb{R}^{2\times 3}$, $\mathcal{W}_{\pm,2}^{2} \in \mathbb{R}^{1\times 3}$, there exist $K' > 0$ and parameter sets $\{\tilde{\mathcal{R}}^{\tilde{q}}, \tilde{\mathcal{R}}_{\pm}^{q}\}_{\tilde{q}\neq\{3,8\}}^{8}$, $\{\tilde{\mathcal{S}}_{\pm}^{q}\}_{q=1}^{7} > 0$ such that

$$\begin{aligned}\|w_1^{\pm}(T) - w_{\pm,1}^{*,1}\| &< \|w_1^{\pm}(0) - \mathcal{W}_{\pm,1}^{1}\| + \tilde{\mathcal{R}}^1 e^{-\tilde{K}T}\\ &+ \tilde{\mathcal{R}}^2 T e^{-\alpha_3 T} + \tilde{\mathcal{R}}_{\pm}^{3} T^2 e^{-T} + \tilde{\mathcal{R}}^4 T^2 e^{-\alpha T} + \tilde{\mathcal{R}}^5 T e^{-n^1 T}\\ &+ \tilde{\mathcal{R}}^6 e^{-\tilde{n}^1 T} + \tilde{\mathcal{R}}^7 e^{-K' T} + \tilde{\mathcal{R}}_{\pm}^{8} e^{-\tilde{\gamma}T} \end{aligned} \tag{15}$$

$$\begin{aligned}\|w_2^{\pm}(T) - w_{\pm,2}^{*,1}\| &< \|w_2^{\pm}(0) - \mathcal{W}_{\pm,2}^{1}\| + \tilde{\mathcal{S}}_{\pm}^{1} e^{-\tilde{K}T}\\ &+ \tilde{\mathcal{S}}_{\pm}^{2} T e^{-\alpha_3 T} + \tilde{\mathcal{S}}_{\pm}^{3} T^2 e^{-T} + \tilde{\mathcal{S}}_{\pm}^{4} T^2 e^{-\alpha T} + \tilde{\mathcal{S}}_{\pm}^{5} T e^{-n^1 T}\\ &+ \tilde{\mathcal{S}}_{\pm}^{6} e^{-\tilde{n}^1 T} + + \tilde{\mathcal{S}}_{\pm}^{7} e^{-\tilde{\gamma}T}. \end{aligned} \tag{16}$$

The estimation parameters set $\{\tilde{\mathcal{R}}^{\tilde{q}}, \tilde{\mathcal{R}}_{\pm}^{q}\}_{\tilde{q}\neq\{3,8\}}^{8}$, $\{\tilde{\mathcal{S}}_{\pm}^{q}\}_{q=1}^{7}$ are linked to outputs y_l, $\Upsilon_{il} \in (0,1)$, initial $\mathcal{W}_2$ and training errors $|e_l| \in [0,1]$. This theorem finishes deducing $\epsilon^1 = \bar{\epsilon}^1 + \eta\theta_{+}^{1}$ at the first iteration based on the vertical propagation pathway.

E. Error Convergence Order

Convergence analysis of algorithms in silicon involves deriving the relationship between the distance ($\|\mathcal{W}^m - \mathcal{W}^{\mathrm{opt}}\|$ or $|\mathcal{E}^m - \mathcal{E}^{\mathrm{opt}}|$) and the iteration number m. From this relationship, the convergence order is determined, reflecting the time required for an algorithm to achieve a specified accuracy level, i.e., the ability to reduce the error distance as m increases.

Until now, we have derived an error estimation for the initial iteration, reasonably described by parameter T, for updating weights in BFCNN compared to that in FCNN. The relationship (15) and (16) ensures that the intrinsic convergence of the algorithm is preserved in the first iteration through our BCRN implementation since the realization error converges regarding T. In addition, it should be noted that the phase length of the chemical oscillator affects not only the runtime of BFCNN but also training accuracy. Then, we give the following technical result to lay the foundation for characterizing the convergence order.

Lemma 7: For arbitrary given positive constant a, b, there must exist $\mathcal{U}, v > 0$ such that $t^a e^{-bt} < \mathcal{U}e^{-vt}$ for $t \geq 0$.

Remark 4: Lemma 7 guarantees that revised parameters $\{\mathcal{U}_{q,\pm}, v_q\}_{q=2}^{5}$, $\{\tilde{\mathcal{U}}_{q,\pm}, \tilde{v}_q\}_{q=2}^{5}$ can be determined for the second to fifth items in (15) and (16), respectively, such that

$$\begin{aligned}\|w_1^{\pm}(T) - w_{\pm,1}^{*,1}\| &< \|w_1^{\pm}(0) - \mathcal{W}_{\pm,1}^{1}\| + \tilde{\mathcal{R}}^{\pm} e^{-vT}\\ \|w_2^{\pm}(T) - w_{\pm,2}^{*,1}\| &< \|w_2^{\pm}(0) - \mathcal{W}_{\pm,2}^{1}\| + \tilde{\mathcal{S}}^{\pm} e^{-\tilde{v}T}\end{aligned}$$

where $\tilde{\mathcal{R}}^{\pm}$ and $\tilde{\mathcal{S}}^{\pm}$ are the sum of revised coefficients in (15) and (16), respectively, and the revised orders $v = \min\{\tilde{K}, \tilde{n}^1, K', \tilde{\gamma}, \{v_q\}_{q=2}^{5}\}$, $\tilde{v} = \min\{\tilde{K}, \tilde{n}^1, \tilde{\gamma}, \{\tilde{v}_q\}_{q=2}^{5}\}$. Due to $\|w_j^{\pm}(0) - \mathcal{W}_{\pm,j}^{1}\| = 0$ for $j = 1, 2$, the initial realization error for training weights is controlled by a negative exponential regarding T with rate $v, \tilde{v}$. In this case, we call the realization error of neural networks through BFCNN possessing *Exponential Order Convergence*, which ensures that a small increment on T will significantly reduce the approximation convergence error. Moreover, larger exponents $v, \tilde{v}$ lead to greater reductions within a fixed time interval. Consequently, the training in BFCNN can converge to the optimal weights within a given accuracy at a relatively fast rate compared to the nonexponential error form.

V. Cumulative Error of the m_{th} $(m \geq 2)$ Iteration

The error estimation for the first iteration illustrates the current error arising from each module and its vertical propagation pathways from the upstream to the downstream modules. The horizontal propagation from previous iterations affects the realization error when BFCNN enters the mth $(m \geq 2)$ iteration (i.e., the mth period of the chemical oscillator). This process may amplify the original coefficients and introduce lower order error entries, leading to increased accumulative error across iterations and modules. This section derives the general upper bound for the error in any mth iteration through recursion.

Given that the continual operation of BFCNN before achieving the preset accuracy makes experimenters unable to calibrate initial concentrations before every training round, the lateral propagation means that endpoint errors derived in the $m-1$th round become initial errors of the corresponding species for the mth $(m \geq 2)$ iteration. Furthermore, the clear-out module [19] forces the concentrations of species $\{I_y^l, I_{p_i}^l\}$ to end up being near one in the first round, namely

$$|i_y^l(2T) - 1| < |1 - i_y^l(T)|e^{-T}$$
$$|i_{p_i}^l(2T) - 1| < |1 - i_{p_i}^l(T)|e^{-T}$$

and for species $X \in \mathcal{S}^f, \tilde{X} \in \mathcal{S}^p$ cleared out, concentrations to follow

$$|x(3T)| < |x(2T)|e^{-T},\ |\tilde{x}(2T)| < |\tilde{x}(T)|e^{-T}$$

where $\mathcal{S}^f = \{P_{il}^{\pm}, Y_l^{\pm}\}$ and $\mathcal{S}^p = \{Y_s^l, Y_e^l, \tilde{Y}^l, P_{is}^l, \tilde{P}^l\}$. Based on the specific initial deviation emerging from $m = 1$, the horizontal propagation pathways can be investigated by spanning from $m = 1$ to $m = 2$. Hereafter, the variable $x^m(t) = x(t + o(m-1)T)$ with o denoting the number of total phases in which species X participate in one period. Now, we showcase that the assignment module preserves convergence orders in subsequent iterations.

Proposition 5: If $c^{*,m} = (\xi_{1+m\tilde{p}-jp}, \ldots, \xi_{(m+1)\tilde{p}-jp})^\top$ $\tilde{c}^{*,m} = \mathbb{O}_{p\times\tilde{p}}$, and $s^{*,m} = (\chi_{\cdot(m-1)\tilde{p}+1-jp}, \ldots, \chi_{\cdot m\tilde{p}-jp})^\top$ with $j = \lfloor\frac{\tilde{p}m}{p}\rfloor$ $(m \nmid \frac{p}{\tilde{p}})$ and $j = \lfloor\frac{\tilde{p}m}{p}\rfloor - 1$ $(m \mid \frac{p}{\tilde{p}})$, and $k > 1$, there exist parameters sequences $m_1^m, m_3^m, \tilde{m}_3^m$ for $m \geq 2$ such that

$$\|s^m(T) - s^{*,m}\| \leq m_1^m e^{-T}$$
$$\|c^m(3T) - c^{*,m}\| \leq m_3^m e^{-kT}$$
$$\|\tilde{c}^m(2T) - \tilde{c}^{*,m}\| \leq \tilde{m}_3^m e^{-kT}. \tag{17}$$

This result implies that the error convergence order of the assignment module is always maintained regardless of iteration numbers as long as the rate constant is designed as $k > 1$. Despite that, a significant impact on the total error estimation of the feedforward module arises from the initial deviation of weights. Next, we show the difference in the general feedforward output errors compared to Proposition 2 and Corollary 1 by deriving the specific estimation formula.

Proposition 6: Using $\{\tilde{\mathcal{R}}_{\pm}^{qe}\}^1, \{\tilde{\mathcal{S}}_{\pm}^{qe}\}^1$ to denote the exponential upper bounds of (15) and (16), letting $\{\tilde{\mathcal{R}}^{qe}\}^1 = \{\tilde{\mathcal{R}}_{+}^{qe}\}^1 + \{\tilde{\mathcal{R}}_{-}^{qe}\}^1, \{\tilde{\mathcal{S}}^{qe}\}^1 = \{\tilde{\mathcal{S}}_{+}^{qe}\}^1 + \{\tilde{\mathcal{S}}_{-}^{qe}\}^1$, abbreviating $\|w_j^{+,m}(0) - \mathcal{W}_{+,j}^m\| + \|w_j^{-,m}(0) - \mathcal{W}_{-,j}^m\|$ as $\tilde{\mathcal{E}}^j(0)_m$ with $j \in \{1, 2\}$, $m = 1$ here, and making $p^{*,2} = \Upsilon^2$, then there exist coefficient sets $\{\tilde{\mathcal{C}}_r^2\}_{r=1}^2$, $\{\mathcal{D}_r^2\}_{r=1}^5$ such that the output error of species $\{P_i^l, Y^l\}$ during $m = 2$ is estimated as

$$\|p^2(T) - p^{*,2}\| \leq \{A\}^2 + \sum_{r=1}^{2} \tilde{\mathcal{C}}_r^2 \left[\{\tilde{\mathcal{R}}^{qe}\}^1 + \tilde{\mathcal{E}}^1(0)_1\right] T^{r-1}$$
$$\|y^2(T) - y^{*,2}\| \leq \{E\}^2 + \sum_{r=1}^{3} \mathcal{D}_r^2 \left[\{\tilde{\mathcal{R}}^{qe}\}^1 + \tilde{\mathcal{E}}^1(0)_1\right] T^{r-1}$$
$$+ \sum_{r=4}^{5} \mathcal{D}_r^2 \left[\{\tilde{\mathcal{S}}^{qe}\}^1 + \tilde{\mathcal{E}}^2(0)_1\right] T^{r-4}. \tag{18}$$

Here, compared to the error bound in the previous iteration of certain species, we call the redundant item (introduced through error propagation) appended to the error bound in the current iteration for the same species as *tail item* (TI), denoted as $(\|x^m(\beta T) - x^{*,m}\|)_{\text{TI}}$, e.g., $(\|y^2(T) - y^{*,2}\|)_{\text{TI}} \triangleq \{\mathcal{D}_y\}^2$ and $(\|p^2(T) - p^{*,2}\|)_{\text{TI}} \triangleq \{\mathcal{D}_p\}^2$ refers to the item except for $\{A\}^2$ and $\{E\}^2$ in Proposition 6, respectively. Following the derivation of Propositions 5 and 6, the horizontal accumulation of errors on initial concentrations of species, excluding weight species, is incapable of introducing redundant entries in the assignment and feedforward modules. Moving forward, as shown in Fig. 4, vertical propagation from $y_l^2(T), p_{il}^2(T), s_3^{l,2}(T)$ and lateral accumulation on initial concentrations of other species have compound effects on pBCRN. Due to the clear-out module designed to prevent the occurrence of lower order than e^{-T} from previous iterations by decaying species or assigning fixed concentrations to species in advance, it can be easily confirmed that TIs in this part solely arise from vertical propagation. Thus, we could derive TIs regarding $\{\tilde{y}_l^2(T), e_l^{\pm,2}(T), s_y^{l,2}(T)\}$ to be $\{\mathcal{D}_y\}^2$ and regarding $\{\tilde{p}_{il}^2(T), s_{p_i}^{l,2}(T)\}$ to be $\{\mathcal{D}_p\}^2$. Consequently, we provide the general error upper bound of updating weights in any mth iteration by recursion, i.e., based on the horizontal propagation pathway, in the main theorem.

Theorem 7: For arbitrary iteration m, we can find the corresponding coefficient sets $\{\mathcal{D}_r^m\}_{r=1}^5$, $\{\tilde{\mathcal{C}}_r^m\}_{r=1}^2$, $\{\mathcal{B}_r^m\}_{r=1}^3$, $\{\mathcal{F}_r^m\}_{r=1}^3$ such that the deviation of updated weights in BFCNN from that in FCNN caused by our realization method is covered by the error upper bound

$$\begin{pmatrix} \tilde{\mathcal{ER}}(T) \\ \tilde{\mathcal{ES}}(T) \end{pmatrix}_m \leq \sum_{j=0}^{m-2} \{ \begin{pmatrix} \tilde{\mathcal{R}} \\ \tilde{\mathcal{S}} \end{pmatrix}_{m-j} \cdot \prod_{i=m-j+1}^{m} ([BFD]_i + \tilde{\mathcal{P}}) \}$$
$$+ \prod_{j=2}^{m} ([BFD]_j + \tilde{\mathcal{P}}) \cdot \begin{pmatrix} \tilde{\mathcal{R}} \\ \tilde{\mathcal{S}} \end{pmatrix}_1 \tag{19}$$

with $\mathcal{E}\tilde{\mathcal{R}}(T)_m = \|w_1^{+,m}(T) - w_1^{-,m}(T) - \mathcal{W}_1^{m+1}\|$, $\mathcal{E}\tilde{\mathcal{S}}(T)_m = \|w_2^{+,m}(T) - w_2^{-,m}(T) - \mathcal{W}_2^{m+1}\|$, and

$$\begin{pmatrix}\tilde{\mathcal{R}}\\ \tilde{\mathcal{S}}\end{pmatrix}_m = \begin{pmatrix}\{\tilde{\mathcal{R}}^{qe}\}^m\\ \{\tilde{\mathcal{S}}^{qe}\}^m\end{pmatrix}, \quad \tilde{\mathcal{P}} = \begin{pmatrix}1 & 2\tilde{p}\\ 0 & 1+2\tilde{p}\end{pmatrix}$$

$$[BFD]_m = 2\tilde{p}\begin{pmatrix}\sum_{r=1}^{3}\mathcal{B}_r^m T^{r-1} & \sum_{r=4}^{5} 5\mathcal{D}_r^m T^{r-4}\\ \sum_{r=1}^{3}\mathcal{F}_r^m T^{r-1} & \sum_{r=4}^{5} 3\mathcal{D}_r^m T^{r-4}\end{pmatrix}.$$

The general error bound of the real updated weights in (19) confirms the convergence of errors regarding T, implying that the individual module design guided by exponential convergence and the method of connecting modules with oscillators for iterative loops are theoretically feasible. The bound explicitly connects the iteration number m, the finite time T, parameters and computational values in FCNN, showing which parameters appear as primary factors in the exponential term and which ones are secondary factors appearing only in the coefficients. The constants and coefficients, while difficult to determine precisely in the current form, are necessary to ensure the generality of the upper bound, allowing it to accommodate various implementation details and system constructions. Since these variables are fixed during any specific training process under a given design, the upper bound provides a quantitative tool to estimate the accuracy of the CRN-based implementation and control errors. It means that practitioners can substitute their predefined m, T and network parameters into the formula to calculate a concrete error threshold before initiating experiments, allowing for adjusting initial concentrations of weighting species, preprocessing "biological input data," and presetting adjustments to meet desired precision requirements, thereby optimizing BFCNN performance and experimental resource allocation.

In addition, the upper bound quantitatively reveals that the upper bound increases over the iteration numbers due to the accumulative components. This can be mitigated by extending the phase length. However, growing the lengthening T results in a longer runtime, presenting a tradeoff between accuracy and efficiency when running BFCNN. Theorem 7 shows that the error bounds for calculations by our design comprise entries with the form of $T^a e^{-bT}$ for $a > 0, b > 0$. It implies that BFCNN exhibits the exponential-order convergence of errors in the finite mth iteration, as indicated by lemma 7, thereby allowing for enhanced precision at a relatively low cost of efficiency. Note that the term $\tilde{p}^{m-1}Me^{-bT}$ appearing in (19) with a potential large $\tilde{p}, m$ will not diverge since the exponential decay term typically decreases much faster than the polynomial growth as T becomes sufficiently large. Furthermore, due to the concrete description of powers outlined in (15) and (16), increasing the magnitude of the parameters in the FCNN within a valid range (without causing time-scale separations between modules) offers an alternative approach to increase the decay rate. Specifically, the decay rate parameters $\alpha, \alpha_3, n,$ and $\tilde{n}$ are influenced by the choice of initial weights, training samples, and the input values $\mathcal{N}^m, \tilde{\mathcal{N}}^m$ of neurons. Similarly, $\tilde{K}$ and K' are related to the initial concentrations and the reaction rate k in the precalculation module. Finally, the reaction rates can be accelerated by increasing the amplitude of the oscillator to achieve a more precise realization.

Ultimately, this upper bound reflects the deviation between the weight species concentrations (i.e., $w(\beta_m T)$ with β_m relying on m) in BFCNN and the desired weight values (i.e., $\mathcal{W}^m$) of FCNN for the current iteration, rather than directly measuring the deviation between the weight concentration sequence (i.e., $\{w(\beta_1 T), \ldots, w(\beta_m T)\}$) and the optimal values (i.e., $\mathcal{W}^{\text{opt}}$) during training. Therefore, it does not imply that the training error for BFCNN (depending on both the realization error and the convergence characterization of FCNN) will increase with increasing m and $\tilde{p}$ even though the upper bound of realization error increases with them due to error accumulation. The significance primarily lies in proving that even though the realization error of BFCNN may increase with several factors, the upper bound remains exponentially convergent with respect to the phase length T. Furthermore, the error control methods discussed above facilitate the management of the realization error perturbations inherent during the training process of BFCNN.

VI. Application to Binary Classification Problems

In this section, we use the same two logical classification problems "OR" and "XOR" as those in Part I for numerical experiments in the Matlab environment to validate the results of Theorems 4 and 7. The basic information about them may be found in Part I, where the constructed BFCNN has been applied to classify them. Here, we conduct further experiments on them to observe the overall trend of the total error upper bound of weight update errors (referred to as "realization errors" henceforth) about phase lengths and iteration numbers in the process of using BFCNN.

By fixing the amplitude of the oscillator as 1, we change the rate constant of oscillatory reactions to produce a group of suitable phase lengths. Then, with different phase lengths BFCNN is executed multiple times on the logical classification problems "OR" and "XOR". The upper bound of realization errors $\|w^{+,m}(T) - \mathcal{W}_+^m\| + \|w^{-,m}(T) - \mathcal{W}_-^m\|$ at each iteration are recorded, where $\mathcal{W}_\pm^m$ are the reference values given in FCNN. The initial weight values for BFCNN are set as the same as those in Part I. Namely, for the "OR" case

$$w^+(0) = \begin{bmatrix}4.1968 & 1.8964 & 2.8458\\ 1.4442 & 1.2924 & 3.0160\\ 7.4030 & 8.3528 & 2.5640\end{bmatrix}, \; w^-(0) = \begin{bmatrix}2 & 6 & 6\\ 2 & 2 & 2\\ 6 & 10 & 2\end{bmatrix}$$

and for the "XOR" case

$$w^+(0) = \begin{bmatrix}3 & 3 & 2.5\\ 4 & 3 & 2\\ 2 & 3 & 2.5\end{bmatrix}, \; w^-(0) = \begin{bmatrix}4 & 4 & 1\\ 3 & 2 & 2.5\\ 1 & 2 & 4\end{bmatrix}.$$

Realization errors change versus phase length T is divided into two cases, as shown in Fig. 6. Since the training was finished after the fifth training round and no weight update occurred from the third to the fourth round when training "OR," we plotted realization error change curves in the second, third, and sixth rounds, respectively. This illustrates the decreasing nature of $\mathcal{E}\tilde{\mathcal{R}}(T)_m, \mathcal{E}\tilde{\mathcal{S}}(T)_m$ with increasing T. For

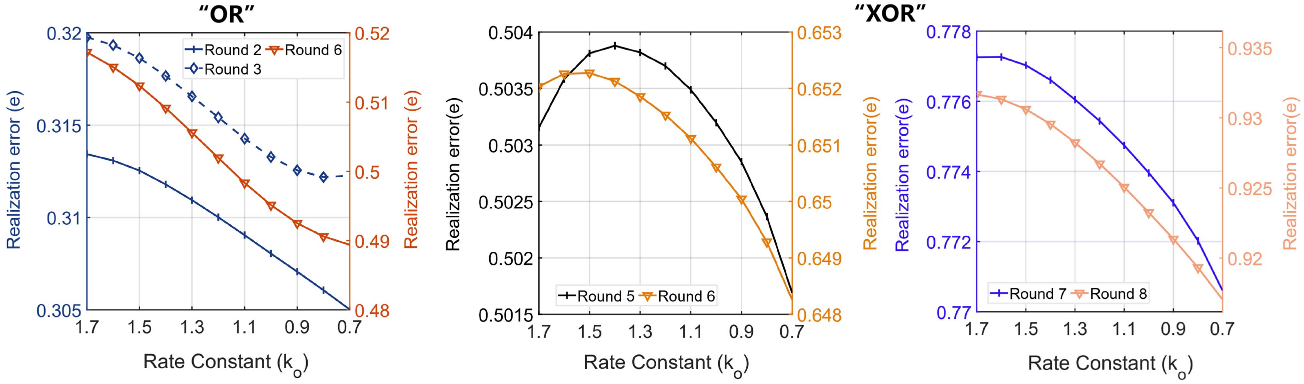

Fig. 6. Realization error change versus the finite phase length from $T = 21.481(k_o = 1.7)$ units to $T = 49.289(k_o = 0.7)$ units of the chemical oscillator. The left picture is for the “OR” case and the right two are for the “XOR” case.

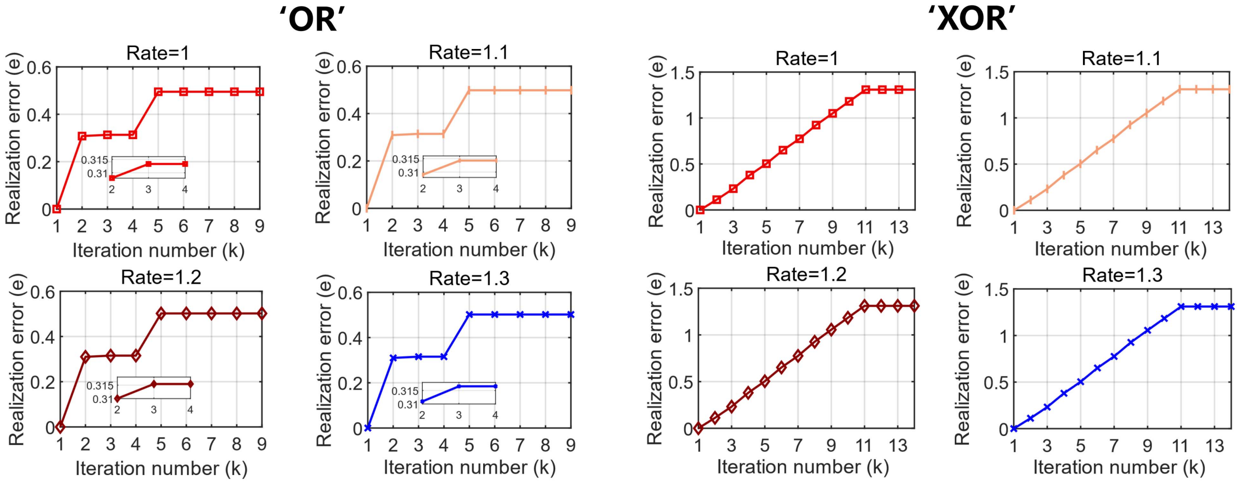

Fig. 7. Realization error change versus the iteration numbers from the first round to the convergence round for different rates k_o. The left picture is for the “OR” case and the right one is for the “XOR” case.

the “XOR” case, we selected four representative training round numbers to show the changing characteristics. Although the curves in the fifth and sixth rounds exhibit overshooting and not monotonically decreasing, the general tendency of realization error is to be reduced without contradicting the bound estimation for realization errors. Furthermore, the error cumulative phenomenon regarding iteration numbers can be observed in Fig. 6, where the y-axis ranges of error curves with larger round numbers always lie above those of smaller round numbers. Finally, more concrete demonstrations are displayed in Fig. 7 that are generated by the error data from the first round to the convergence round under certain phase lengths, confirming cumulative incremental trends. As can be seen from this figure, the realization error for the “OR” case increases in steps while the “XOR” case increases smoothly. The reason is that for the former there is almost no change in the realization error from the 3rd to the 4th iteration while it increases over other iterations. For example, at Rate $= 1.2$, the realization errors are $[1st, 4 \times 10^{-13}]$, $[2nd,\ 0.310017]$, $[3rd,\ 0.315425]$, $[4th,\ 0.315428]$, $[5th,\ 0.501947]$, $[6th,\ 0.501953]$, and so on. This means that no weight update occurs from the 3rd to the 4th iteration, which leads to almost no change in the realization error.

VII. Conclusion and Future Studies

Part II has constructed a comprehensive framework for analyzing errors of BFCNN introduced in Part I, arising from representing computing results utilizing the finite approximation of LSSs, along with their vertical propagation and horizontal accumulation. In particular, this approach derives the general error upper bound of training weights in each iteration by examining errors in the first iteration to clarify the vertical propagation pathway and combining the influence from previous iterations to identify the horizontal propagation pathway. Our framework addresses a crucial gap in considering the impact of the occurrence and propagation of LSSs approximation errors on the performance of biological circuits designed by BCRN realization, which has the potential to offer effective ways to reduce errors.

Also, it should be noted that there are no technique difficulties in extending the current work to other deeper networks. The main reason is that the BFCNN is designed by considering all information processing parts of artificial neural networks, i.e., the input step of the training samples, the feedforward computation, the weight update, the termination of the training, and the possibility of multiple training iterations in Part I. Specifically, we designed a biochemical assignment module for inputting training samples of small batches in sequence, the judgment module for terminating training based on error precision, and used the chemical oscillator that can perform multiple iterations. These three biochemical information processing modules are undoubtedly directly applicable in any biochemical neural network or machine learning algorithm. For the feedforward computation in FCNN with Sigmoid activation function, we have devised the lwsBCRN and sigBCRN that separately handle linear computations and activation function computation at neural nodes. More lwsBCRN and sigBCRN units should be added and connected by the oscillator when increasing the depth and width of BFCNN. As for the weight update by gradient descent-based backpropagation, we designed the ngBCRN and uBCRN that aim to compute the negative gradient of the mean square loss function and new weights for new iterations, respectively. The core of constructing ngBCRN is the developed split factor multiplication strategy for computing the backpropagation formula of each weight increment, which ensures that whenever the depth of FCNN is increased, it is only necessary to increase the number of reactions according to this strategy to calculate additional factors compared to the back layer. Moreover, the mechanism of uBCRN can be directly integrated into other algorithms to calculate the sum of the old weights and the weight increments, thereby obtaining the new weights. The framework for the error analysis established in Part II follows the same logic, which can also be applied to more general scenarios. Current error describes the error source caused by the widely used realization method. Accumulative error characterizes the error propagation along the vertical direction (caused by the cascade of modules via chemical oscillators in one iteration) and horizontal direction (resulting from the multiple iterations necessary in any algorithm) through biased parameters and initial concentrations, leading to the deviation of desired LSSs. In addition, the error analysis of the assignment module (Proposition 1), lwsBCRN and sigBCRN in the feedforward module (Lemma 1, Proposition 2), as well as ngBCRN constructed based on the SPF strategy in the learning module (Lemma 6, Proposition 3), can be directly applied to more general biochemical neural networks built using these modules. The error analysis process will become more complex accordingly.

Accumulative error defined in this article characterizes the discrepancy between current LSSs and desired LSSs, revealing that error propagation can cause deviations in the target set point of a module after multiple iterations. Essentially, this represents a disturbance to the steady state of biological circuits induced by the realization errors studied here, leading to the loss of the robust perfect adaptation. This is a critical challenge in synthetic biology, which can be addressed by designing integral controllers to enhance robustness against various environmental disturbances or uncertainties, thereby restoring the adaptation of biological circuits [25], [26]. Therefore, a promising avenue for further research concerns designing integral controllers for computational modules to conquer disturbances from accumulative errors, ensuring that the LSS is maintained at a fixed level. In addition, inaccuracies caused by nonzero concentrations in the "turn OFF" phase of oscillators should be appended to establish the new error format in the future.

References

[1] Y. Qian, C. McBride, and D. Del Vecchio, "Programming cells to work for us," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 411–440, 2018.

[2] Y. Qian and D. Del Vecchio, "Robustness of networked systems to unintended interactions with application to engineered genetic circuits," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 4, pp. 1705–1716, Dec. 2021.

[3] S. K. Aoki, G. Lillacci, A. Gupta, A. Baumschlager, D. Schweingruber, and M. Khammash, "A universal biomolecular integral feedback controller for robust perfect adaptation," *Nature*, vol. 570, no. 7762, pp. 533–537, 2019.

[4] T. Frei, C.-H. Chang, M. Filo, A. Arampatzis, and M. Khammash, "A genetic mammalian proportional–integral feedback control circuit for robust and precise gene regulation," *Proc. Nat. Acad. Sci.*, vol. 119, 2022, Art. no. e2122132119.

[5] L. Qian, E. Winfree, and J. Bruck, "Neural network computation with DNA strand displacement cascades," *Nature*, vol. 475, no. 7356, pp. 368–372, 2011.

[6] K. M. Cherry and L. Qian, "Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks," *Nature*, vol. 559, no. 7714, pp. 370–376, 2018.

[7] C. Zhang et al., "Cancer diagnosis with dna molecular computation," *Nature Nanotechnol.*, vol. 15, no. 8, pp. 709–715, 2020.

[8] D. Soloveichik, G. Seelig, and E. Winfree, "DNA as a universal substrate for chemical kinetics," *Proc. Nat. Acad. Sci.*, vol. 107, no. 12, pp. 5393–5398, 2010.

[9] F. Fages, G. Le Guludec, O. Bournez, and A. Pouly, "Strong turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs," in *Proc. Comput. Methods Syst. Biol.: 15th Int. Conf.*, 2017, pp. 108–127.

[10] M. Vasić, D. Soloveichik, and S. Khurshid, "CRN : Molecular programming language," *Natural Comput.*, vol. 19, no. 2, pp. 391–407, 2020.

[11] M. Vasić, C. Chalk, A. Luchsinger, S. Khurshid, and D. Soloveichik, "Programming and training rate-independent chemical reaction networks," *Proc. Nat. Acad. Sci.*, vol. 119, no. 24, 2022, Art. no. e2111552119.

[12] M. Vasic, C. Chalk, S. Khurshid, and D. Soloveichik, "Deep molecular programming: A natural implementation of binary-weight RELU neural networks," in *Proc. PMLR Int. Conf. Mach. Learn.*, 2020, pp. 9701–9711.

[13] C. C. Samaniego, A. Moorman, G. Giordano, and E. Franco, "Signaling-based neural networks for cellular computation," in *Proc. 2021 IEEE Amer. Control Conf.*, 2021, pp. 1883–1890.

[14] D. Arredondo and M. R. Lakin, "Supervised learning in a multilayer, nonlinear chemical neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7734–7745, Oct. 2023.

[15] M. R. Lakin, "Design and simulation of a multilayer chemical neural network that learns via backpropagation," *Artif. Life*, vol. 29, no. 3, pp. 308–335, 2023.

[16] D. F. Anderson, B. Joshi, and A. Deshpande, "On reaction network implementations of neural networks," *J. Roy. Soc. Interface*, vol. 18, no. 177, 2021, Art. no. 20210031.

[17] D. Blount, P. Banda, C. Teuscher, and D. Stefanovic, "Feedforward chemical neural network: An in silico chemical system that learns xor," *Artif. Life*, vol. 23, no. 3, pp. 295–317, 2017.

[18] Y. Fan, X. Zhang, and C. Gao, "Towards programming adaptive linear neural networks through chemical reaction networks," *IFAC-PapersOnLine*, vol. 55, no. 18, pp. 7–13, 2022.

[19] Y. Fan, X. Zhang, C. Gao, and D. Dochain, "Automatic implementation of neural networks through reaction networks–Part I: Circuit design and convergence analysis," *IEEE Trans. Automat. Control*, early access, Mar. 24, 2025, doi: 10.1109/TAC.2025.3554428.

[20] M. Feinberg, "Complex balancing in general kinetic systems," *Arch. Rational Mechan. Anal.*, vol. 49, no. 3, pp. 187–194, 1972.

[21] M. Feinberg, “Lectures on chemical reaction networks,” *Notes Lectures Given Math. Res. Center, Univ. Wisconsin*, vol. 49, 1979, doi: 10.5281/zenodo.10631900.

[22] F. Horn and R. Jackson, “General mass action kinetics,” *Arch. Rational Mechan. Anal.*, vol. 47, no. 2, pp. 81–116, 1972.

[23] M. Lachmann and G. Sella, “The computationally complete ant colony: Global coordination in a system with no hierarchy,” in *Proc. Eur. Conf. Artif. Life*, 1995, pp. 784–800.

[24] M. Banaji, B. Boros, and J. Hofbauer, “The smallest bimolecular mass-action system with a vertical Andronov–Hopf bifurcation,” *Appl. Math. Lett.*, vol. 143, 2023, Art. no. 108671.

[25] C. Briat, A. Gupta, and M. Khammash, “Antithetic integral feedback ensures robust perfect adaptation in noisy biomolecular networks,” *Cell Syst.*, vol. 2, no. 1, pp. 15–26, 2016.

[26] Y. Qian and D. Del Vecchio, “Realizing ‘integral control’ in living cells: How to overcome leaky integration due to dilution?,” *J. Roy. Soc. Interface*, vol. 15, no. 139, 2018, Art. no. 20170902.

Yuzhen Fan received the B.Sc. degree in information and computational science from the College of Science, Northeastern University, Shenyang, China, in 2020. She is currently working toward the Ph.D. degree in operation research and cybernetics with the School of Mathematical Sciences, Zhejiang University, Hangzhou, China.

Her research interests include control theory, synthetic biology, and chemical reaction networks theory.

Xiaoyu Zhang received the B.Sc. degree in mathematics from Anhui University, Hefei, China, in 2017, and the Ph.D. degree in operation research and cybernetics from Zhejiang University, Hangzhou, China, in 2022.

She is currently an Associate Researcher with the School of Mathematics, Southeast University, Nanjing, China. Her research focuses on control and chemical reaction networks theory.

Chuanhou Gao (Senior Member, IEEE) received the B.Sc. degree in chemical engineering from the Zhejiang University of Technology, Hangzhou, China, in 1998, and the Ph.D. degree in operational research and cybernetics from Zhejiang University, Hangzhou, in 2004.

From June 2004 until May 2006, he was a Postdoc with the Department of Control Science and Engineering, Zhejiang University. Since June 2006, he has joined the Department of Mathematics, Zhejiang University, where he is currently a Professor. From October 2011 to October 2012, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include synthetic biology from the viewpoint of control, chemical reaction network theory, machine learning, and mixed integer programming with applications.

Dr. Gao is an Associate Editor for IEEE Transactions on Automatic Control and *International Journal of Adaptive Control and Signal Processing*.

Denis Dochain (Fellow, IEEE) received the master degree in electrical engineering and the dual joint Ph.D. degree in applied science with the “thèse d'agrégation de l'enseignement supérieur' from the Université Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, in 1982, 1986, and 1994, respectively.

He has been CNRS Associate Researcher with the LAAS, Toulouse, France, in 1989, and Professor with the Ecole Polytechnique de Montréal, Montreal, Canada, in 1987–88 and 1990–92. Since 1990, he has been with the FNRS, Brussels, Belgium. Since September 1999, he is a Professor with the ICTEAM (Institute), Université Catholique de Louvain, and Honorary Research Director of the FNRS. Between 2002 and 2004 he was invited Professor with Queen's University, Kingston, ON, Canada. Since 2005, he has been a Full Professor with the UCL, London, U.K. He is the (co-)author of 5 books, more than 160 papers in refereed journals, and 260 international conference papers. His main research interests include in the field of nonlinear systems, thermodynamics-based control, parameter and state estimation, adaptive extremum seeking control and distributed parameter systems, with application to microbial ecology, environmental, biological and chemical systems, and electrical and mechanical systems.

Dr. Dochain is the Editor-in-Chief of the *Journal of Process Control*, Senior Editor of the IEEE Transactions of Automatic Control, and Associate Editor of *Automatica*. He is active in IFAC since 1999 (Council member, Technical Board member, Publication Committee Member and chair, TC and CC Chair). He was the recipient of the IFAC outstanding service award in 2008 and is an IFAC Fellow since 2010. He was the recipient of the title of Doctor Honoris Causa from the INP Grenoble on December 13, 2020.