2024 Tenth Indian Control Conference (ICC)
9 - 11 December, 2024
IISER Bhopal, India

Physics Informed Deep Operator Networks for Baculovirus-Insect Cell System**

Vishnu Swaroopji Masampally*, Surbhi Sharma*, Lopamudra Giri*, and Kishalay Mitra*

Abstract— **The Baculovirus expression vector system (BEVS) is extensively used for producing recombinant proteins, virus-like particles (VLPs), and vaccines. Optimizing and controlling this process requires a robust model tuned with experimental data. A physics-based mathematical model for BEVS has been developed and validated through experiments. This model has been used to create a Physics-Informed Deep Operator Network (PI-DeepONet) model. Given that the mathematical model involves stiff ordinary differential equations (ODEs), the Stiff-PINN formulation, a variant of Physics-Informed Neural Networks (PINNs), is utilized to construct the loss function. The PI-DeepONet model effectively predicts process dynamics across various initial conditions, whereas the Stiff-PINN model is limited to the initial condition for which it was trained. Once developed, PI-DeepONet models can replace standard PINN models, enabling the optimization and control of BEVS across diverse initial conditions.**

I. Introduction

The COVID-19 pandemic underscored an unprecedented demand for vaccines. Controlling global diseases and eradicating numerous infections, such as those caused by the novel coronavirus, require large-scale vaccine development [1]. There are significant challenges posed by the differences in upstream production scales between laboratory and industrial settings. To address these challenges, the co-development of bioreactor technology and computational biology is essential. This integration is now more important than ever [2].

The Baculovirus expression vector system (BEVS) is extensively used for development of recombinant proteins, virus-like particles (VLPs), and vaccines [3, 4]. However, this process is marked with batch-to-batch unpredictability, low yields, and poor product quality due to costly and time-consuming trial-and-error experimentation. Therefore, predictive mathematical models and optimization frameworks are necessary to increase efficiency, reduce manufacturing costs, and alleviate these challenges while meeting current demands. The necessary mathematical and optimization frameworks of the Baculovirus-insect cell system (BICS) should be both accurate and robust.

The predictive models are classified into three types: physics-based, mechanistic and hybrid. Physics-based models rely on fundamental physical principles to describe and predict BICS dynamics using equations derived from the first principles. Mechanistic models focus on capturing detailed processes and interactions within systems, often representing biological pathways or chemical reactions with mathematical formulations. Data-driven models, in contrast, leverage empirical data to generate predictions based on statistical patterns and machine learning algorithms [5].

Although data-driven models have shown great potential in modeling complex biological systems, these models fail to incorporate fundamental governing equations of the system. This results in inaccuracies, especially in scenarios where understanding underlying mechanisms is crucial [6, 7]. Mechanistic models, on the other hand, provide detailed insights into biological processes by explicitly modeling interactions and pathways. However, they may oversimplify or ignore certain complex interactions and dependencies, limiting their predictive capability in real-world applications. Therefore, hybrid modeling, that combines the strengths of both mechanistic and data-driven modeling is a possible practical approach [8].

Hybrid modeling for process systems began in the early 1990s, with bioprocess hybrid modeling starting in the late 1990s and continuing to evolve with more accurate results [5, 9 – 16]. Recent advancements in Physics-Informed Neural Networks (PINNs) integrate physics-based models into neural networks, enhancing computational efficiency and robustness [17 - 19]. PINNs use prior physics knowledge to guide learning, making them effective for modeling complex systems like the Baculovirus-insect cell system (BICS), where data collection is challenging and costly.

Alireza et al. introduced a framework for physics-informed machine learning in biosystems, noting accurate predictions for some variables but challenges in predicting all variables [2]. The impact of stiff equations on PINN accuracy is explored in [20]. They further proposed a method to reduce stiffness by assuming quasi-steady-state (QSS) conditions, altering the ODEs. De Florio et al. developed a single-layer neural network integrating PINNs with extreme theory of functional connections to handle stiff ODEs effectively without QSS assumptions [21].

Despite extensive study in various fields, PINNs' application in biosystems remains limited [22 – 24]. These limitations further include difficulties in accurately capturing system dynamics and dependencies at various initial or boundary conditions. To address these challenges, Physics-informed Deep Operator Network (PI-DeepONet) represents a promising advancement, integrating deep operator networks with physics-informed learning to enhance predictive accuracy and accommodate diverse system conditions effectively for complex systems such as BICS.

**Research supported by Department of BioTechnology, GOI (BT/PR34209/AI/133/19/2019).

Vishnu Swaroopji Masampally, Surbhi Sharma, Lopamudra Giri, and Kishalay Mitra are with Department of Chemical Engineering, Indian Institute of Technology, Hyderabad, Telangana, 502284 INDIA (*corresponding author. phone: 040-2301-7016; e-mail: kishalay@che.iith.ac.in)

In this study, PI-DeepONet is applied to model the BICS. To address the stiffness in the governing equations, the Stiff-PINN formulation is selected from among multiple variants of PINNs designed for stiff ordinary differential equations (ODEs). The effectiveness of PI-DeepONet is evaluated using data from the experiment. Section 2 provides a detailed discussion of the experimental procedures, mathematical modeling of BICS, and the methodology employed by PI-DeepONet. Section 3 presents the results of mathematical model validation and assesses the accuracy of PI-DeepONet. Conclusions drawn from this study are outlined in Section 4.

TABLE I. THE MATHEMATICAL MODEL FOR THE BACULOVIRUS – INSECT SYSTEM

Variable	Equation	Description
	$\mu = \mu_{max}\frac{[S]}{K_s+[S]}\frac{[O_2]}{K_{O_2}+[O_2]}\frac{K}{\exp\left(\frac{[CO_2]}{K_{CO2}}\right)+1}$ $\mu_1 = \mu_{max}\frac{[S]}{K_s+[S]}$	specific growth rate induced by $[O_2]$ and inhibited by $[CO_2]$
Substrate, [S]	$\frac{d[S]}{dt} = -\mu_1\frac{[U]+[I]}{[Y_s]}$	Rate of consumption of [S] by [U] and [I]
Oxygen, $[O_2]$	$\frac{d[O_2]}{dt} = k_L a(0^* - [O_2]) - \frac{[U]+[I]}{[Y_{O_2}]}$	Rate of change of $[O_2]$ due to mass transfer and uptake by [U] and [I]
Uninfected cells, [U]	$\frac{d[U]}{dt} = \mu[U] - a_1[V][U] - \frac{k_{d1}[U]}{\exp(10[S])}$	Rate of generation of [U] due to growth of [U], [V] attachment to [U] and death of [U]
Infected cells, [I]	$\frac{d[I]}{dt} = a_1[V][U] - k_d[I]$	Rate of generation of [I] due to attachment of [V] to [U] and death of the [I]
Dead Cells, [D]	$\frac{d[D]}{dt} = \frac{k_{d1}[U]}{\exp(10[S])} + k_d[I]$	Rate of generation of [D] due to dead [U] and dead [I]
Virus, [V]	$\frac{d[V]}{dt} = k_a[I] - k_{dv}[V]$	Rate of production of [V] from [I] and decay of [V]
Carbon-di-oxide $[CO_2]$	$\frac{d[CO_2]}{dt} = \mu\frac{[U]+[I]}{[Y_{CO_2}]}$	$[CO_2]$ produced from [U] and [I]
	$Cell\ denity = U + I$ $Total\ cell\ denity = U + I + D$	Viable cell density and total cell density

II. MATERIALS AND METHODS

A. Experiments

Spodoptera frugiperda (Sf-9) insect cells were cultured in 250 mL shaker flasks containing 100 mL of nutrient-rich media. These cells were infected with the wild type baculovirus (AcMNPV) in a batch suspension culture. Viable cell density (VCD) and percentage viability were measured at 24-hour intervals from the start of the experiment up to 96 hours. Data were collected in triplicate for both VCD and percentage viability. Further details on viral strains, culture conditions, and experimental procedures can be found in the literature [25].

B. Mathematical Model

The mathematical model for the BICS developed and validated in our previous work is employed in this study [25]. This model, composed of coupled ODEs, describes the growth dynamics of insect cells infected with baculovirus. In addition to cell growth, it accounts for the dynamics of substrate concentration, oxygen concentration, carbon dioxide concentration, and virus levels. Table 1 presents the model for BICS. These coupled ODEs are solved using the Explicit Runge-Kutta (RK) method of order 5(4) as an initial value problem.

C. Physics-Informed Neural Networks (PINNs) and Stiff-PINNs

PINNs are a type of deep neural network (DNN) that utilize automatic differentiation to compute derivatives of neural network outputs with respect to their inputs and model parameters [17]. The DNN takes independent variables as inputs and predicts the dependent variables as output variables. Training of this DNN involves minimizing the mean squared error (MSE) of the governing equations, and thereby ensuring that the neural network adheres to these equations. PINNs can learn various governing equations, including algebraic equations, ODEs, time-independent partial differential equations (PDEs), time-dependent PDEs, integro-differential equations, and fractional PDEs.

To handle the stiffness in the ODEs, the Stiff-PINN approach compares the timescales of dependent variables and assumes quasi-steady-state (QSS) conditions for variables with shorter timescales. By assuming QSS, the rate of change of these variables is set to zero, converting their governing ODEs into algebraic equations. QSS variables are approximated using algebraic equations within the DNN loss function. This reduces the number of ODEs and the number of DNN outputs. In the BICS as described in Table 1, oxygen concentration varies an order of magnitude lower than the other variables [24]. Therefore, oxygen concentration is considered as a QSS variable in this system and the algebraic equation for oxygen concentration is as derived in (1). The modified system consists of 6 ODEs with an analytical expression for oxygen concentration.

$$[O_2] = \left(0^* - \frac{[U]+[I]}{[Y_{O_2}]}10^6\right)\Big/ k_L a \tag{1}$$

D. Physics-Informed Deep Operator Network (PI-DeepONet) for Baculovirus-Insect Cell System (BICS)

PI-DeepONet, is a sophisticated approach designed to learn and approximate the operator mapping from input functions to output functions in complex dynamical systems. This framework leverages both physics-based models and deep learning, enhancing the ability to predict system behavior under various initial conditions. In this work, we create a PI-DeepONet model, that acts as a robust surrogate for the BICS as described in Table 1.

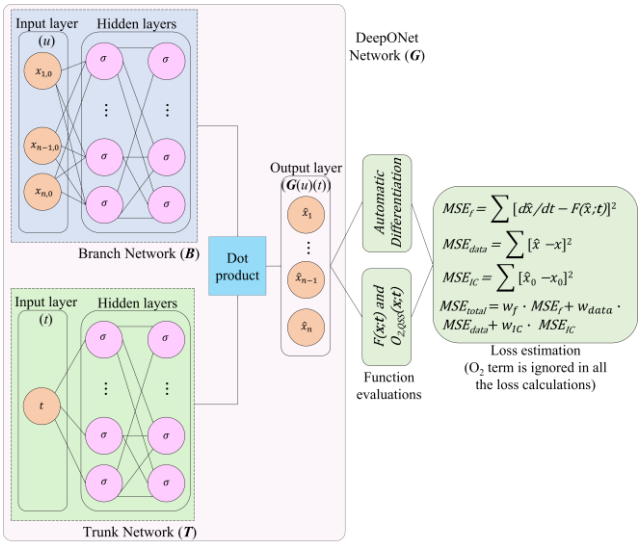


Figure 1: Schematic of the PI-DeepONet where the initial conditions u is input to the branch network and time t is the input to the trunk network and the outputs from the DeepONet network G(u)(t) as explained in Table 1.

PI-DeepONet consists of two main components: a branch network and a trunk network. The branch network processes the input function values, whereas the trunk network handles the input locations where the operator is evaluated. Given a function $u(t)$, the objective of PI-DeepONet is to learn an operator G such that: $G(u)(t) \approx v(t)$, where $v(t)$ represents the output function dependent on $u(t)$. The branch network takes samples from $u(t)$ at various points as inputs, while the trunk network takes corresponding locations t as inputs. The outputs of these networks are then combined to approximate the operator G.

The branch network, $\boldsymbol{B}$, and the trunk network, $\boldsymbol{T}$, can be mathematically represented as: $\boldsymbol{B} = \{\boldsymbol{B_1}, \boldsymbol{B_2}, \ldots, \boldsymbol{B_m}\}$ and $\boldsymbol{T} = \{\boldsymbol{T_1}, \boldsymbol{T_2}, \ldots, \boldsymbol{T_m}\}$, where m is the number of sensors. For a set of sampled inputs $u = \{u(t_1), u(t_2), \ldots, u(t_n)\}$, the branch network produces a feature vector: $\boldsymbol{B_i} = \boldsymbol{B}(u(t_i))$. Similarly, for a set of locations $t = \{t_1, t_2, \ldots, t_n\}$, the trunk network produces another feature vector: $\boldsymbol{T_j} = \boldsymbol{T}(t_j)$. The final output of PI-DeepONet at location t is then a combination of these feature vectors: $\boldsymbol{G}(u)(t) \approx \sum \boldsymbol{B}_i \cdot \boldsymbol{T}_i$.

In the context of BICS, the input function u to the branch network ($\boldsymbol{B}$) is assumed to provide the initial condition, sampled only at $t = 0$. Consequently, the PI-DeepONet model delivers the required outputs for various initial conditions by feeding these initial conditions into the branch network ($\boldsymbol{B}$) and the time input into the trunk network ($\boldsymbol{T}$). The coupled ODEs in Table 1 and the PI-DeepONet network are represented by (2) and (3).

$$d\boldsymbol{x}/dt = F(\boldsymbol{x};t) \tag{2}$$

$$\boldsymbol{x}_\theta(t) = \boldsymbol{G}_\theta(u)(t) \tag{3}$$

where $\boldsymbol{x}$ = [[S], [U], [I], [D], [V], [CO_2]], $F(\boldsymbol{x};t)$ is the vector of RHS of the equations in Table 1. A detailed architecture of PI-DeepONet model for $F(\boldsymbol{x}; t)$ is shown in Figure 1. The model predictions from $\boldsymbol{G}$ are finetuned using network parameters θ. The predictions from equation (3), $\boldsymbol{x}_\theta(t)$, are as explained in the equation $\boldsymbol{G}(u)(t) \approx \sum \boldsymbol{B}_i \cdot \boldsymbol{T}_i$ where u is initial condition sampled at $t = 0$ and t is time input. The loss function for training PI-DeepONet model is designed to minimize the discrepancy between the predicted output and the true output while incorporating the physics-informed constraints. The total loss, $MSE(\boldsymbol{\theta})$, is as given in (4).

$$MSE(\boldsymbol{\theta}) = w_f \cdot MSE_f(\boldsymbol{\theta}) + w_{data} \cdot MSE_{data}(\boldsymbol{\theta}) + w_{IC} \cdot MSE_{IC}(\boldsymbol{\theta}) \tag{4}$$

The total loss function, $MSE(\boldsymbol{\theta})$, of the PI-DeepONet comprises three components: the mean squared error of the coupled ODE system $MSE_f(\boldsymbol{\theta})$ multiplied with weight w_f with N_f collocation points, the mean squared error from the N_u sampled data points $MSE_{data}(\boldsymbol{\theta})$ multiplied with weight w_{data}, and the mean squared error associated with the initial conditions $MSE_{IC}(\boldsymbol{\theta})$ multiplied with weight w_{IC} over N_{IC} data points. Both $MSE_{data}(\boldsymbol{\theta})$ and $MSE_{IC}(\boldsymbol{\theta})$ are supervised losses because they are calculated using the numerical solution. In contrast, $MSE_f(\boldsymbol{\theta})$ is an unsupervised loss, enabling the PI-DeepONet to learn the physics of the problem without the data. The three components of the total loss are detailed in equations (5)-(7).

$$MSE_f(\boldsymbol{\theta}) = (1/N_f)\Sigma\,[d\boldsymbol{G}_\theta(u)(t)/dt - F(\boldsymbol{x}_\theta;\boldsymbol{t})]^2 \tag{5}$$

$$MSE_{data}(\boldsymbol{\theta}) = (1/N_u)\Sigma[\,\boldsymbol{G}_\theta(u)(t) - \boldsymbol{x}]^2 \tag{6}$$

$$MSE_{IC}(\boldsymbol{\theta}) = (1/N_{IC})\Sigma[\,\boldsymbol{G}_\theta(u)(t)_{t=0} - \boldsymbol{x}_{t=0}]^2 \tag{7}$$

E. Computational requirements

The models are trained with a processor of base speed 1.80 GHz, 10 cores, 12 logical processors and a RAM of 32 GB.

III. Results and Discussion

In machine and deep learning, variable scaling significantly impacts model accuracy. For the BICS mathematical model, where variables have different magnitudes, adjusting the units to million cells (M cells) or using non-dimensionalized equations helps improve training accuracy and model performance. The scaling of BICS is discussed in detail in [24]. The scaled model's validation is performed against experimental data for VCD and % viability of BICS [24].

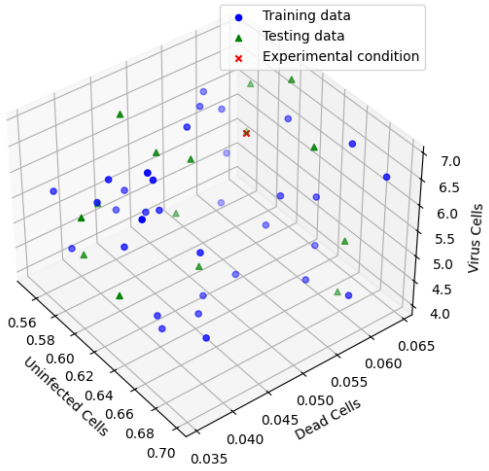


Figure 2: Figure showing various initial conditions used for training, testing and validation of PI-DeepONet model

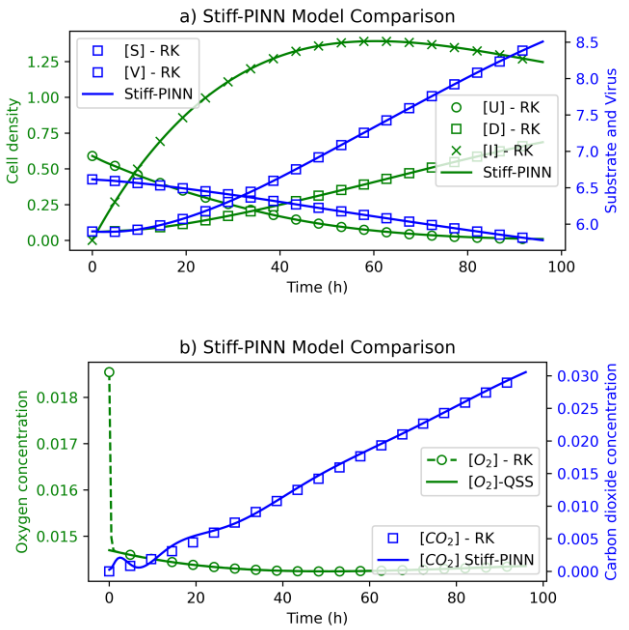


Figure 3: Performance of Stiff-PINN results compared with numerical solution obtained using numerical solution.

A. Dataset

For this study, a comprehensive dataset was generated to train and validate the PI-DeepONet model. The dataset comprises simulations from 50 different initial conditions, with 200 data points collected at each initial condition. To ensure robust training and testing, the dataset was split with a 70:30 ratio, where 70% of the data, i.e., data from 35 initial conditions was used for training and 30% of the data, i.e., data from 15 initial conditions, was used for testing. The data for 50 various initial concentrations are generated by varying few key variables while keeping other variables constant across all the initial conditions. The initial condition for substrate concentration (6.616), oxygen concentration (0.0185), carbon-dioxide concentration (0.0) and infected cell concentration (0.0) are kept constant across all the initial conditions. The initial conditions of uninfected cell density, dead cell density and virus cells are randomly selected from a uniform distribution between [0.55 – 0.7], [0.035 – 0.065] and [4 – 7] units respectively. Figure 2 depicts the distribution of initial conditions generated for train and test datasets for uninfected cells, dead cells and virus cells. The PI-DeepONet model is further validated for an experimental condition for a known initial condition.

B. Physics Informed Neural Networks

A detailed study on effectiveness of Stiff-PINN for BICS is demonstrated in [24]. This Stiff-PINN model uses a fully connected neural network with an input layer (with 1 neuron), three hidden layers (with 40 neurons each), and one output layer, (6 neurons). Tan sigmoid (tanh) is the activation function for all the neurons in this network. The performance of the Stiff-PINNs at an initial condition for which the model is trained is as shown in Figure 3. The initial condition is same as the Eperiment-1 shown in Figure 2. The results in Figure 3 are further used to compare PI-DeepONet model predictions subsequently.

C. Physics Informed Deep Operator Network (PI-DeepONet)

1) Network architecture

The input layer of the branch network (***B***) is set to have 7 neurons corresponding to the number of variables in the initial condition vector for the BICS. The input layer of the trunk network (***T***) is set to 1 neuron, representing the time variable, *t*. The output layer consists of 6 neurons, reflecting the number of state variables predicted by the model, excluding the QSS species, oxygen concentration. The model includes 3 hidden layers, with 32 neurons each, for ***B*** and ***T*** each, providing the depth required to model complex relationships within the BICS data. This PI-DeepONet model architecture consists of 6854 network parameters.

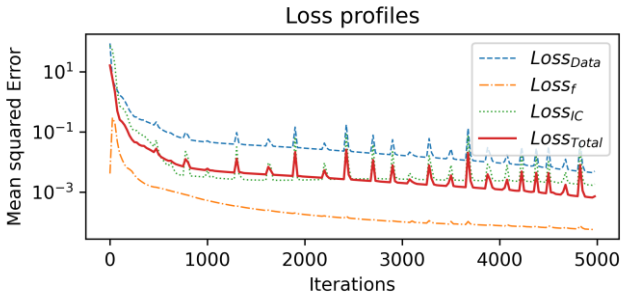


Figure 4: Evolution of total loss and three components of the total loss while training PI-DeepONet.

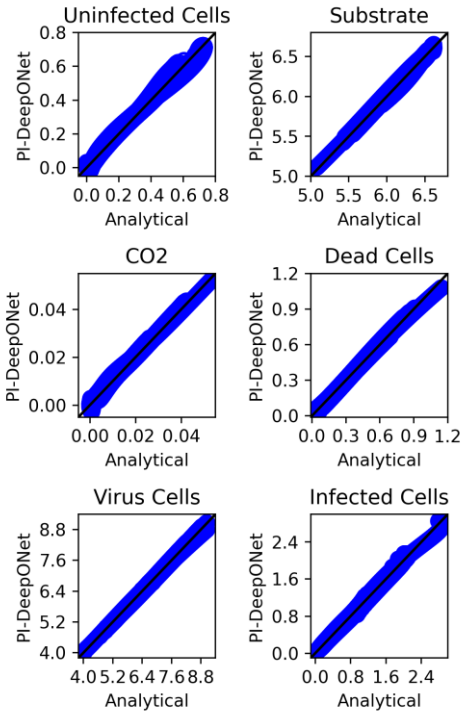


Figure 5: Parity plot of predictions at various initial conditions from test dataset using PI-DeepONet

2) Model training

The data with 35 initial conditions for training and 15 initial conditions for testing, with 200 time points in each solution resulted in 7000 data points for training and 3000

data points for testing. The 7000 data points are used for estimating the three components of the total loss function, $MSE_f(\boldsymbol{\theta})$, $MSE_{IC}(\boldsymbol{\theta})$ and $MSE_{data}(\boldsymbol{\theta})$. The weights of function loss, data loss and initial condition loss are set to $w_f = 0.8$, $w_{IC} = 0.8$ and $w_{data} = 0.1$. Adam optimization algorithm with a learning rate of 0.001 is used to estimate parameters of the PI-DeepONet model. This model is trained for 5,000 iterations and with a total MSE of 0.00268. The profiles of the losses of each component and the overall losses are plotted in Figure 4 with the y-axis being plotted on log-scale for better visibility of convergence for each loss component.

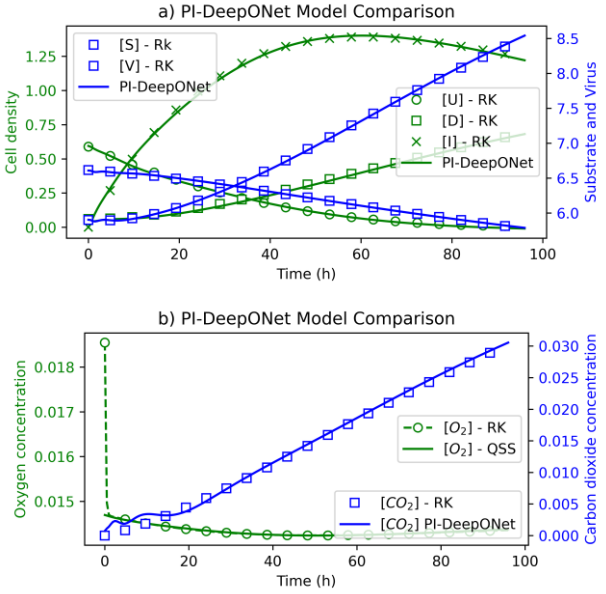


Figure 6: Performance of PI-DeepONet results compared with numerical solution for an initial condition of Experiment-1 in Figure 2.

3) Model predictions

The predictions from the PI-DeepONet are compared with the numerical results obtained from mathematical model. Figure 5 displays the parity plots of all the predictions from the 15 different initial conditions in the test data set. It is observed from Figure 5 that the predictions are satisfactory for all the initial conditions in the test data. One initial condition from the experiments is further used to validate the PI-DeepONet model and the predictions are as shown in Figure 6. It is observed from Figure 6a and Figure 6b that the PI-DeepONet model predicts the dynamic behavior of the cell concentrations, substrate concentration, oxygen and carbon dioxide concentrations accurately compared to the numerical solutions of both original ODEs and the modified QSS ODEs. The PI-DeepONet predictions in Figure 6 are similar to that of prediction from Stiff-PINN model in Figure 3 that was trained for this one initial condition.

Since the PI-DeepONet can capture the state dynamics accurately for various initial conditions, this model can further be used for optimization and control of bioreactor. Therefore, the generalization capabilities of PI-DeepONet over PINNs gives it an edge in the optimization and control framework. Further, PI-DeepONet model can overcome instabilities in numerical solutions while solving stiff ODEs with limited data of as less than 35 initial conditions data.

D. Future work

In this work, the design of the neural network architecture or the choice of hyper-parameters, including the weights of the loss functions used during the training of PI-DeepONets is based on trial-and-error studies. In future work, we plan to perform an ablation study that provides insights on the influence of design architecture and wights on the model performance. Similarly, the amount of data also affects the model performance. Therefore, in the future work, we plan to understand the effect of the number of initial conditions in the data and the data points per each condition on the model's accuracy and generalizability.

IV. Conclusion

A mathematical model for BICS has been validated using experimental data. A deep neural network model based on physics-informed neural networks (PINNs) was trained for stiff equations in BICS. However, Stiff-PINNs lacked generalization across different initial conditions. To address this, PI-DeepONet, a variant of the deep operator network designed to handle multiple initial conditions, was employed. This PI-DeepONet effectively captured BICS dynamics across 35 different initial conditions, demonstrating accurate performance in these scenarios.

Acknowledgment

K. Mitra and L. Giri acknowledge the support provided by the project BT/PR34209/AI/133/19/2019 funded by the Department of Biotechnology (DBT), Government of India, for this work. The authors thank Mr. K. Ananth Krishnan, Dr. Venkataramana Runkana, Tata Consultancy Services (TCS) for their encouragement and guidance.

References

[1] S. Pagliusi et al., "Vaccines: Shaping global health," *Vaccine*, vol. 35, no. 12, pp. 1579–1585, 2017. doi:10.1016/j.vaccine.2017.02.017

[2] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, *Systems Biology informed Deep Learning for inferring parameters and hidden dynamics*, 2019. doi:10.1101/865063

[3] A. Contreras-Gómez, A. Sánchez-Mirón, F. García-Camacho, E. Molina-Grima, and Y. Chisti, "Protein production using the baculovirus-insect cell expression system," *Biotechnology Progress*, vol. 30, no. 1, pp. 1–18, 2013. doi:10.1002/btpr.1842

[4] B. Szewczyk, L. Hoyos-Carvajal, M. Paluszek, I. Skrzecz, and M. Lobo de Souza, "Baculoviruses — re-emerging biopesticides," *Biotechnology Advances*, vol. 24, no. 2, pp. 143–160, 2006. doi:10.1016/j.biotechadv.2005.09.001

[5] V. S. Masampally, A. Pareek, and V. Runkana, "Cascade Gaussian process regression framework for biomass prediction in a fed-batch reactor," *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018. doi:10.1109/ssci.2018.8628937

[6] M. S. Bangi, A. Narasingam, P. Siddhamshetty, and J. S.-I. Kwon, "Enlarging the domain of attraction of the local dynamic mode decomposition with control technique: Application to hydraulic fracturing," *Industrial & Engineering Chemistry Research*, vol. 58, no. 14, pp. 5588–5601, 2019. doi:10.1021/acs.iecr.8b05995

[7] B. Bhadriraju, M. S. Bangi, A. Narasingam, and J. S. Kwon, "Operable adaptive sparse identification of systems: Application to chemical processes," *AIChE Journal*, vol. 66, no. 11, 2020. doi:10.1002/aic.16980

[8] M. S. Bangi and J. S.-I. Kwon, "Deep Hybrid Modeling of Chemical Process: Application to hydraulic fracturing," *Computers & Chemical*

Engineering, vol. 134, p. 106696, 2020. doi:10.1016/j.compchemeng.2019.106696

[9] D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992. doi:10.1002/aic.690381003

[10] H. T. Su and T. J. McAvoy, "Integration of multilayer perceptron networks and Linear Dynamic models: A hammerstein modeling approach," *Industrial & Engineering Chemistry Research*, vol. 32, no. 9, pp. 1927–1936, 1993. doi:10.1021/ie00021a017

[11] J. Schubert, R. Simutis, M. Dors, I. Havlik, and A. Lübbert, "Bioprocess optimization and Control: Application of hybrid modelling," *Journal of Biotechnology*, vol. 35, no. 1, pp. 51–68, 1994. doi:10.1016/0168-1656(94)90189-9

[12] M. L. Thompson and M. A. Kramer, "Modeling chemical processes using prior knowledge and neural networks," *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994. doi:10.1002/aic.690400806

[13] H. Preusting, J. Noordover, R. Simutis, and A. Lübbert, "The use of hybrid modelling for the optimization of the penicillin fermentation process," *CHIMIA*, vol. 50, no. 9, p. 416, 1996. doi:10.2533/chimia.1996.416

[14] M. S. Bangi, A. Narasingam, P. Siddhamshetty, and J. S.-I. Kwon, "Enlarging the domain of attraction of the local dynamic mode decomposition with control technique: Application to hydraulic fracturing," Industrial & Engineering Chemistry Research, vol. 58, no. 14, pp. 5588–5601, 2019. doi:10.1021/acs.iecr.8b05995

[15] M. S. Bangi, A. Narasingam, P. Siddhamshetty, and J. S.-I. Kwon, "Enlarging the domain of attraction of the local dynamic mode decomposition with control technique: Application to hydraulic fracturing," Industrial & Engineering Chemistry Research, vol. 58, no. 14, pp. 5588–5601, 2019. doi:10.1021/acs.iecr.8b05995

[16] B. Bayer, R. Dalmau Diaz, M. Melcher, G. Striedner, and M. Duerkop, "Digital Twin Application for model-based DOE to rapidly identify ideal process conditions for space-time yield optimization," *Processes*, vol. 9, no. 7, p. 1109, 2021. doi:10.3390/pr9071109

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed Neural Networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. doi:10.1016/j.jcp.2018.10.045

[18] G. E. Karniadakis et al., "Physics-informed Machine Learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021. doi:10.1038/s42254-021-00314-5

[19] V. Jadhav, A. Deodhar, A. Gupta, and V. Runkana, "Physics informed Neural Network for health monitoring of an Air Preheater," *PHM Society European Conference*, vol. 7, no. 1, pp. 219–230, 2022. doi:10.36001/phme.2022.v7i1.3343

[20] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, "Stiff-pinn: Physics-informed neural network for stiff chemical kinetics," *The Journal of Physical Chemistry A*, vol. 125, no. 36, pp. 8098–8106, 2021. doi:10.1021/acs.jpca.1c05102

[21] M. De Florio, E. Schiassi, and R. Furfaro, "Physics-informed Neural Networks and functional interpolation for stiff chemical kinetics," Chaos: *An Interdisciplinary Journal of Nonlinear Science*, vol. 32, no. 6, p. 063107, 2022. doi:10.1063/5.0086649

[22] M. S. Bangi, K. Kao, and J. S.-I. Kwon, "Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β-carotene production using saccharomyces cerevisiae," *Chemical Engineering Research and Design*, vol. 179, pp. 415–423, 2022. doi:10.1016/j.cherd.2022.01.041

[23] Tipireddy, Ramakrishna, Paris Perdikaris, Panos Stinis, and Alexandre Tartakovsky. "A comparative study of physics-informed neural network models for learning unknown dynamics and constitutive relations." arXiv preprint *arXiv:1904.04058 (2019)*

[24] V. S. Masampally, S. Sharma, L. Giri, and K. Mitra, "Physics Informed Neural Networks for Baculovirus-Insect Cell System," Dec. 2023.

[25] S. Sharma, P. N. Keerthi, L. Giri, and K. Mitra, "Toward performance improvement of a baculovirus–insect cell system under uncertain environment: A robust multiobjective dynamic optimization approach for Semibatch Suspension Culture," *Industrial & Engineering Chemistry Research*, vol. 62, no. 1, pp. 111–125, 2022. doi:10.1021/acs.iecr.2c03355