

# Enhancing purple non-sulfur bacteria modeling with physics-informed neural networks

Matheus C. R. Nunes\* Laurent Dewasme\* Manon Gilson\*\*  
Guillaume Bayon-Vicente\*\* Baptiste Leroy\*\* Alain Vande Wouwer\*

\* *Systems, Estimation, Control and Optimization (SECO), University of Mons,  
Belgium (e-mail: matheus.camargoromanonunes@umons.ac.be,  
laurent.dewasme@umons.ac.be, alain.vandewouwer@umons.ac.be).*

\*\* *Laboratory of Proteomics and Microbiology, University of Mons, Belgium  
(e-mail: manon.gilson@umons.ac.be,  
guillaume.bayon-vicente@umons.ac.be, baptiste.leroy@umons.ac.be).*

---

**Abstract:** In the pursuit of a resource-efficient economy, purple non-sulfur bacteria (PNSB) represent a promising solution due to their capacity to convert waste from various sources into valuable products, including biomass. However, scaling up PNSB technology remains challenging, and developing reliable dynamic models for monitoring and control is essential to facilitate this transition. Despite recent efforts dedicated to PNSB modeling, existing gaps in process understanding and difficulties in data collection still limit their development. In this regard, physics-informed neural networks (PINNs) emerge as a natural candidate, considering their ability to integrate partial physical information. This paper presents a PINN-based model developed by combining an existing first-principles model—whose performance declines under new conditions—with additional data representing these conditions. To assess the performance of the PINN-PNSB model, we compare it with other modeling alternatives, including the updated parametric model obtained by classical parameter identification and a pure artificial neural network (ANN). A PINN-derived model, obtained by updating the physical model parameters during PINN training, is also evaluated. Results using training and test data demonstrate the superior performance of the PINN-based model for PNSB applications.

*Keywords:* physics-informed neural networks, hybrid modeling, machine learning, parameter estimation, biotechnology.

---

## 1. INTRODUCTION

Despite their recent introduction, physics-informed neural networks (PINNs) (Raissi et al., 2019) have quickly established a leading role among modern modeling techniques. The key aspect of PINNs lies in their ability to learn systems governed by ordinary differential equations (ODEs) and partial differential equations (PDEs) by incorporating knowledge provided by the physical model during the training phase. Through automatic differentiation (AD), one can effectively compute the derivatives of the state variables predictions with respect to the inputs (often time and space coordinates). Thus, the physical model acts as a soft constraint during hyper-parameter optimization. Like traditional artificial neural networks (ANNs), one or multiple initial conditions, as well as experimental data, can also be added. As discussed in (Karniadakis et al., 2021), in contrast to pure black-box methods, the incorporation of physical information significantly alleviates the requirements for data acquisition. Although adding experimental data is optional, it becomes mandatory for inverse problems, where the goal is to update the physical model parameters during PINN training simultaneously.

The potential to work with small datasets is particularly appealing for bioprocess applications, where data collection is usually a difficult task. In this context, a few PINN-based applications have been proposed. Rogers et al. (2023b) developed a shallow PINN to predict biomass growth and substrate consumption in a

fermentation process. In addition, the authors identify the time-constant and time-varying parameters of the physical model. In (Bangi et al., 2022), the PINN framework is associated with neural ordinary differential equations (neural ODEs) to accelerate training and construct a model for  $\beta$ -carotene production by *S. cerevisiae*. The resulting hybrid model shows superior accuracy when compared to an existing kinetic model. Biologically-informed neural networks (BINNs) are introduced by Lagergren et al. (2020) and applied to predict cell migration in *in-vitro* experiments, showing superior accuracy when compared to classical physical models. Several other PINN and non-PINN hybrid models applied to bioprocesses can be found in the literature, for instance, (Rogers et al., 2023a; Cui et al., 2024).

Regarding purple non-sulfur bacteria (PNSB), PINNs can serve as a valuable tool as producing experimental data is often challenging. In this regard, Hülsen et al. (2022) refer to the scarcity of data from large-scale PNSB systems as one of the drawbacks preventing PNSB technology from progressing. Furthermore, physical phenomena governing PNSB growth and substrate interactions are only partially understood and subject to ongoing research, making PNSB systems ideal candidates for PINN-based applications. In these cases, a PINN-PNSB model can be designed to connect partially known physics and limited experimental data.

In a previous study (Nunes et al., 2024), we developed a mechanistic model to predict the growth of the PNSB strain

*Rs. rubrum* on a mixture of monosaccharides. Despite the model satisfactory predictive performance, it exhibits a high parametric variability. As the model limitations become more evident once new conditions are introduced, we address this drawback by proposing a PINN-PNSB model that combines the existing model with a small dataset reflecting the new conditions. This strategy takes advantage of the PINN ability to integrate known physics captured by the mechanistic model and unknown dynamics learned from experimental data.

This paper is organized as follows: Section 2 presents the physical model and its limitations. Section 3 details the PINN design, training, and simulation. Section 4 discusses the numerical results regarding the PINN-based model and compares its performance to other modeling approaches. Finally, conclusions and perspectives are drawn in Section 5.

## 2. PROBLEM STATEMENT

The first-principles model to predict the growth of *Rs. rubrum* cultures with fructose and glucose as carbon sources is described in (1) to (7):



$$\Phi_{fru} = \mu_{\max_{fru}} \frac{fru}{K_{fru} + fru} \quad (3)$$

$$\Phi_{glu} = \mu_{\max_{glu}} \frac{glu}{K_{glu} + glu} \quad (4)$$

$$\frac{1}{X} \frac{dX}{dt} = \Phi_{fru} + \Phi_{glu} \quad (5)$$

$$\frac{1}{X} \frac{dfru}{dt} = -Y_{fru} \Phi_{fru} \quad (6)$$

$$\frac{1}{X} \frac{dglu}{dt} = -Y_{glu} \Phi_{glu} \quad (7)$$

where (1) and (2) represent the macroscopic reactions, (3) and (4) define the Monod-like kinetics, and (5) to (7) are the mass-balance ODEs.

The concentrations of biomass, fructose, and glucose are respectively represented by  $X$ ,  $fru$ , and  $glu$ , all measured in g/L. The model parameters are:  $Y_{fru}$  and  $Y_{glu}$  the yield coefficients,  $\mu_{\max_{fru}}$  and  $\mu_{\max_{glu}}$  the maximum specific growth rates,  $K_{fru}$  and  $K_{glu}$  the half-saturation constants for  $fru$  and  $glu$ , respectively. The corresponding values and their 95% confidence intervals (CI) are shown in Tab.1.

The model successfully predicts PNSB growth on mixtures of fructose and glucose. However, acquiring supplementary experimental data reveals its inability to generalize when glucose is the sole carbon source. Figure 1 shows that the model overestimates biomass growth and glucose consumption after 100 h. The fit deterioration is quantified through the root mean squared error (RMSE) shown in Tab. 2. The first column contains RMSE results for the original dataset described in (Nunes et al., 2024), composed of four batch experiments, and the second column corresponds to the new experiment from Fig. 1. While one could simply revise the model by updating the parameter values using the new dataset, this strategy may not capture the underlying mechanisms that make the system behave differently when PNSB grow on fructose and glucose mixtures versus glucose only. Therefore, we propose to represent the system by

a hybrid PINN-PNSB model. Through this approach, we expect to leverage the physical knowledge embedded in the original model while accounting for the partially understood phenomena through the machine-learning capabilities of PINNs.

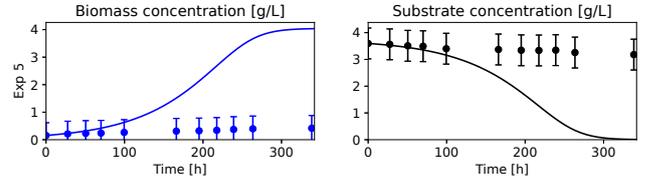


Fig. 1. Experimental measurements (dots) and model predictions for the concentrations of biomass (blue), and glucose (black). The vertical bars correspond to *a posteriori* calculations of the 95% confidence intervals related to the measurement error.

Table 1. Parameter identification results and confidence intervals for the original model.

Parameter	Original model	
	Nominal value	CI (%)
$\mu_{\max_{fru}}$ (1/h)	0.0838	636
$\mu_{\max_{glu}}$ (1/h)	0.0198	95.0
$K_{fru}$ (g/L)	0.181	4514
$K_{glu}$ (g/L)	1.40	150
$Y_{fru}$ (g/g)	2.37	31.0
$Y_{glu}$ (g/g)	0.925	17.0

Table 2. RMSE results for biomass, fructose, and glucose for the original data (biomass cultivation on fructose and glucose) and new data (biomass cultivation exclusively on glucose).

	Original dataset	New dataset
$X$ (g/L)	0.125	1.08
$fru$ (g/L)	0.0133	-
$glu$ (g/L)	0.123	0.932

## 3. PHYSICS-INFORMED NEURAL NETWORKS

### 3.1 Loss function

A scheme detailing the PINN design considered in this work is presented in Fig. 2. The PINN structure consists of two intersecting regions. The first (A) is comparable to a classical feedforward neural network whose inputs are the vector of time measurements  $t$ , and the initial conditions regarding the system states,  $x = [X, fru, glu]$ , represented by  $x_0 = [X_0, fru_0, glu_0]$ , where  $X_0$ ,  $fru_0$  and  $glu_0$  represent the initial concentrations of biomass, fructose, and glucose, respectively. In the intersection between (A) and (B), we find the PINN predictions for the state concentrations  $\hat{x} = [\hat{X}, \hat{fru}, \hat{glu}]$  so that  $\hat{x} = PINN(x_0, t)$  and  $\hat{x}_0 = PINN(x_0, t = 0)$ . In the second region (B), automatic differentiation is applied to compute the time derivatives of the state variables,  $d\hat{x}/dt = [d\hat{X}/dt, d\hat{fru}/dt, d\hat{glu}/dt]$  for biomass, and both substrates. The first term of the loss function (8) is then constructed by computing the difference between those derivatives and the ones from the physical model, represented by  $f(\hat{x}, t, \theta)$ , where  $\theta$  represents the parameter set of the physical model. The satisfaction of the initial conditions is also sought and is added to the loss function via equation (9). Since this study also aims to solve an inverse problem, experimental

measurements  $x_d$  are incorporated into the loss function, as in (10).

$$L_c(w, b, \theta) = \sum_{i=1}^{N_c} (d\hat{x}_i/dt - f(\hat{x}_i, t_i, \theta))^2 \quad (8)$$

$$L_{ic}(w, b, \theta) = \sum_{i=1}^{N_{ic}} (\hat{x}_{0,i} - x_{0,i})^2 \quad (9)$$

$$L_d(w, b, \theta) = \sum_{i=1}^{N_d} (\hat{x}_i - x_{d,i})^2 \quad (10)$$

where the trainable weights and biases are respectively  $w$  and  $b$ .

The losses regarding the physical knowledge  $L_c$ , initial conditions  $L_{ic}$ , and experimental data  $L_d$  are weighted as we multiply them by their respective loss weights  $w_c$ ,  $w_{ic}$ , and  $w_d$ . Finally, the losses are combined to produce a unique loss function as follows:

$$L(w, b, \theta) = w_c L_c + w_{ic} L_{ic} + w_d L_d \quad (11)$$

Tuning loss weights and the number of collocation points  $N_c$  is often done by trial and error. To increase the efficiency of this task, we explore relatively simple structures for PINN-PNSB model, prioritizing the use of compact and fast-to-solve configurations, reducing the effort to a few trials.  $N_c$  is chosen such that the average time difference between consecutive collocation points does not exceed five hours. Setting  $N_c = 80$  collocation points is enough to capture the physical model dynamic during training. However, the selection of weight values is more complex and will be detailed in subsection 3.3. Lastly, determining the number of initial conditions  $N_{ic}$  and data points  $N_d$  is straightforward as it depends exclusively on the datasets used during training.

### 3.2 Definition of the PINN architecture

The PINN-based model was trained by employing the Python library *DeepXDE* (Lu et al., 2021) with the *TensorFlow* (Abadi et al., 2016) backend. The optimal PINN architecture was determined by exploring configurations that balance model reliability and training duration, aiming to keep the training time within minutes. To achieve this, a fully connected feedforward neural network, or multi-layer perceptron (MLP), was constrained to a single hidden layer with a hyperbolic tangent activation function, and various configurations were evaluated by adjusting the number of neurons,  $N$ . Each candidate architecture was trained using data from 5 batch experiments and  $N_c = 80$  collocation points for 40,000 iterations of the Adam (Adaptive Moment Estimation) optimizer with the learning rate  $lr = 5 \times 10^{-4}$ . Initially, two simple structures,  $N = 5$ , and  $N = 10$ , were tested. Subsequently, the number of neurons was increased by increments of 10 until either a significant increase in training time or no improvement in the final loss was observed.

Table 3 presents the results for the tested architectures of the shallow PINN-PNSB model. Notably, an increase in the residual loss occurs at  $N = 30$ . Hence, the selected architecture for the single hidden layer corresponds to  $N = 20$ . Even though all proposed structures demonstrate satisfactory predictive performance, this architecture represents the lowest final loss among the tested configurations. Given that the computation times are

considerably low, this architecture offers an optimal balance between computational efficiency and predictive accuracy.

Although the resulting PINN-based models are already acceptable at this stage, the primary objective is to evaluate the candidate architectures. Therefore, further improvements to the final loss will be performed during the training phase.

Table 3. PINN architecture candidates.

Number of neurons	5	10	20	30
Training time (s)	50	49	53	64
Final loss	476	116	80	267

### 3.3 PINN training and test

The training phase is a two-stage procedure inspired by Rogers et al. (2023b). In this work, the authors have empirically identified that such an approach can be more effective in avoiding undesired local minima.

Loss function minimization is achieved by using its partial derivatives with respect to the trainable parameters. An optimization algorithm, such as Adam or L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno), is employed to solve the optimization problem. Before training starts, the set of nominal values  $\theta$  of the original model is provided and optimized alongside  $w$  and  $b$ . As a result, the model parameters are updated, generating a PINN-derived model. The optimization problem is formulated as follows:

$$\{w^*, b^*, \theta^*\} = \min_{\{w, b, \theta\}} L(w, b, \theta) \quad (12)$$

where  $w^*$  and  $b^*$  are, respectively, the optimal sets of the PINN-PNSB weights and biases, and  $\theta^*$  is the optimal parameter set of the PINN-derived model.

A full-batch optimization is conducted by incorporating data from five independent experiments from a batch reactor (totalizing 130 measurements),  $x_d = [x_{d1}, x_{d2}, x_{d3}, x_{d4}, x_{d5}]$  and  $x_0 = [x_{01}, x_{02}, x_{03}, x_{04}, x_{05}]$ , along with physics from the original model,  $N_c = 80$ . In the first stage, the PINN is trained without physical information ( $w_c = 0$ ,  $w_{ic} = 1$ ,  $w_d = 1$ ) for 20,000 iterations of Adam ( $lr = 5 \times 10^{-4}$ ). The resulting PINN is then trained with  $w_c = 1$ ,  $w_{ic} = 10$ ,  $w_d = 1 \times 10^3$  for 20,000 iterations of Adam with the same learning rate as the previous step, followed by 30,000 iterations of L-BFGS.

To verify the effectiveness of the two-stage training approach, a one-stage training is also performed, consisting of 40,000 iterations of Adam followed by up to 30,000 iterations of L-BFGS, using the same weight configuration as in the second stage of the two-stage optimization. Each training strategy was repeated five times, and the configuration yielding the lowest loss was selected. The resulting PINN-based model is then tested against unseen data, and its predictive performance is evaluated by calculating the RMSE values for each state.

### 3.4 Hybrid vs. classical modeling

To assess the hybrid PINN-PNSB model, we compare its performance to other modeling approaches: 1) the updated physical model obtained through a traditional identification process, 2) a pure black-box artificial neural network (ANN), and 3) the PINN-derived model resulting from the parameter re-identification during PINN training. The model performance

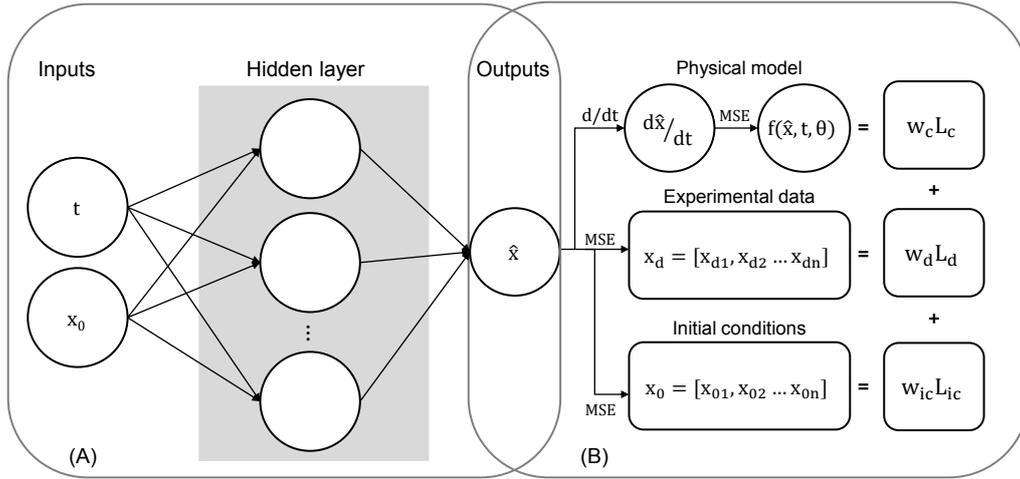


Fig. 2. PINN-PNSB scheme.

analysis consists of evaluating all models in terms of fit quality on both seen and unseen data, respectively, experiments 1-5 and 6, which is achieved by computing the RMSE values. The procedure for model training/identification and testing is detailed in Fig. 3.

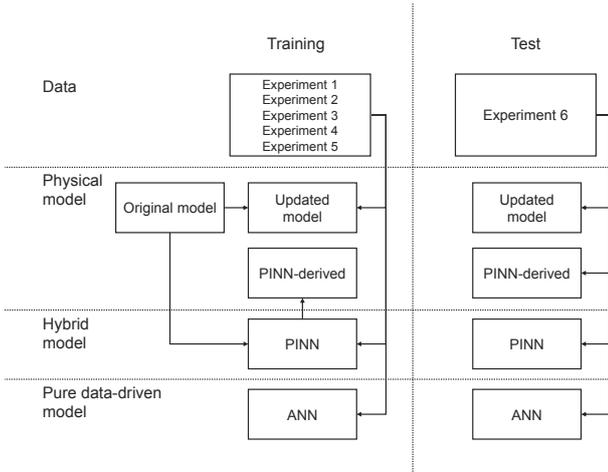


Fig. 3. Schematic representation of the procedure for model comparison.

The methodology for updating the physical model parameters and confidence intervals is based on the procedure described in (Fekih-Salem et al., 2019). Parameters are estimated by minimizing a weighted least-squares criterion, which measures the distance between the model-simulated data and the observed values. The parameter values from the original model are provided to the solver to initialize the optimization process.

The classical ANN architecture and training procedure replicates those of the PINNs, with the key distinction that no physics is incorporated in the form of ODEs. Instead, we enhance the training process by incorporating 10 datasets containing 8 data points for each state, directly generated from simulations of the original model, covering all initial conditions. The generated datasets provide a total of 80 additional training points, aligning with the number of collocation points used in

the PINN training. ANN training is completed in 72 seconds with a residual loss of  $9.3 \times 10^{-3}$ .

## 4. NUMERICAL RESULTS

### 4.1 PINN training and test

Training for both two-stage and single-stage strategies is completed in under five minutes, producing smooth profiles with no significant overfitting. Figure 4 depicts the evolution of the weighted losses. As expected, the two-stage PINN exhibits a spike after 20,000 iterations, corresponding to the inclusion of the model loss at the start of the second stage. However, the loss subsequently decreases and stabilizes at a lower final value than in single-stage training. The average residual losses, computed across five training runs, demonstrate a clear advantage of the two-stage approach, resulting in reductions of 25.8%, 16.1%, and 18.8% in the model, data, and total losses, respectively.

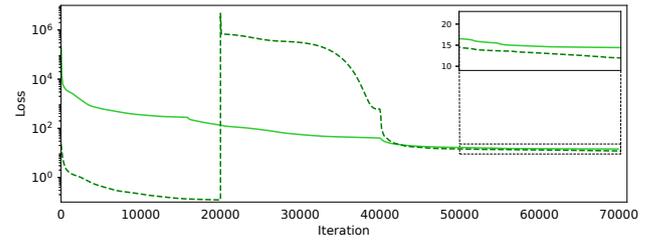


Fig. 4. Loss evolution for the best run of the two-stage (dashed line) and single-stage training (continuous line).

### 4.2 Results of hybrid vs. classical modeling

In this section, we present the PINN-PNSB model performance compared to the other modeling alternatives. Figure 5 presents the resulting PINN and ANN predictions for the state evolutions across five training experiments and one test dataset. Results demonstrate that the proposed PINN-based model effectively predicts biomass growth and substrate consumption for all datasets used during training while successfully performing on unseen data. The training datasets show the full capacity

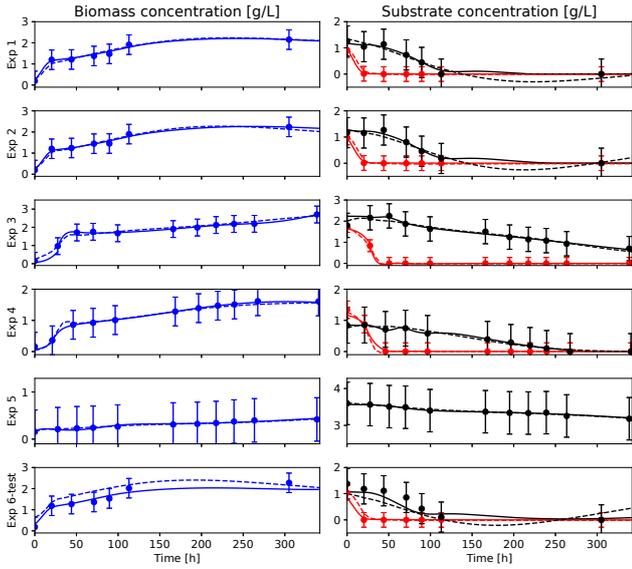


Fig. 5. Experimental measurements (dots) and model predictions for the concentrations of biomass (blue), fructose (red), and glucose (black). The PINN-PNSB predictions are indicated by continuous lines, and the classical ANN predictions are shown in dashed lines. The bars correspond to *a posteriori* calculations of the 95% confidence intervals related to the measurement error.

of PINNs because, in addition to the limited number of measurements, we observe significant gaps between consecutive measurements, notably the last two points of experiments 1 and 2. For this reason, incorporating physical knowledge is crucial to avoid deviations from the expected behavior in regions where no experimental information is provided.

It can also be noted in Fig. 5 that the pure ANN presents unexpected behavior in regions of large intervals between consecutive measurements. This result demonstrates that simply augmenting the training samples by generating data with the physical model is ineffective in this case. In contrast, incorporating physical information during training prevents this and other related issues, such as the transgression of physical constraints (e.g., negative values of biomass and substrate concentrations), which we observe in the classical ANN results but not in the PINN. It is important to mention that the ANN model could be further improved by incorporating additional data or adjusting the architecture and activation function. However, such modifications deviate from the purpose of evaluating the PNSB models under comparable conditions. Additionally, considering the limited complexity of the PNSB system’s dynamics, the MLP—recognized as a universal approximator—represents a reasonable choice.

Figure 6 shows the state predictions of the PINN-derived and classical models, and the respective parameter values and 95% confidence intervals are listed in Table 4. While both parametric models provide an improved fit to the experimental data as compared to the original model, the PINN-derived model demonstrates superior overall performance, especially on the test dataset. However, concerning the confidence interval results, the updated physical model exhibits lower parametric uncertainties than the PINN-derived one. Regardless, both alternatives represent an improvement in parameter uncertainty with

respect to the original PNSB model, reducing the confidence interval for 5 out of 6 parameters.

Despite producing a better fit, parameter identification of the PINN-derived model can be particularly challenging since the parameter values may diverge from the original ones and even assume negative values, losing physical meaning. This issue can be avoided by constraining parameter outputs during optimization, which can be achieved by modifying the *TensorFlow* variable *tf.variable* to force the solution of the optimization problem to be bounded. To accomplish this, we employ a function whose outputs are limited to a certain range, such as the sigmoid.

To draw a rigorous comparison of all considered methods, we compute the RMSE values for training and test datasets, as indicated in Tab. 5 and 6, respectively. Results highlight the PINN superior performance for most tested scenarios, with the only exception being observed for fructose. However, this state may be considered less representative than the others since it is rapidly depleted, leaving only a few non-zero measurements for evaluation. Interestingly, the classical ANN outperforms the updated physical and PINN-derived models on the training datasets. However, this outcome is not replicated on unseen data, where the PINN-derived model surpasses it. This result is expected as the classical ANN tends to overfit the data, while for the PINN, the embedded physics constrains such behavior. Consequently, the hybrid model generalizes better than the pure data-driven approach when applied to the studied case. Overall, both qualitative and quantitative analysis confirm the PINN potential to improve the modeling of PNSB cultures, showing it can successfully address issues imposed by process understanding and data limitations.

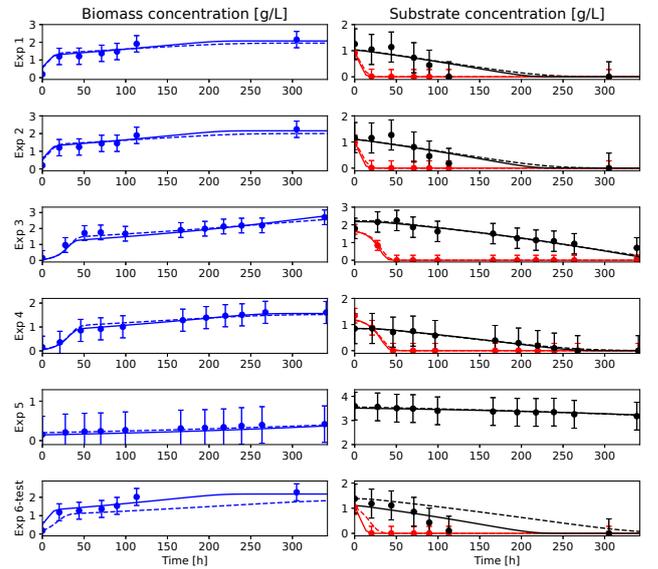


Fig. 6. Experimental measurements (dots) and model predictions for the concentrations of biomass (blue), fructose (red), and glucose (black). The updated physical model is indicated by dashed lines, and the PINN-derived model is shown in continuous lines. The bars correspond to *a posteriori* calculations of the 95% confidence intervals related to the measurement error.

Table 4. Parameter identification results for the updated physical model and PINN-derived model.

Parameter	Updated model		PINN-derived model	
	Value	CI (%)	Value	CI (%)
$\mu_{\max_{fru}}$ (1/h)	0.125	60.0	0.108	128
$\mu_{\max_{glu}}$ (1/h)	0.0021	29.0	0.0028	21.0
$K_{fru}$ (g/L)	0.619	159	0.389	537
$K_{glu}$ (g/L)	0.140	218	0.0574	399
$Y_{fru}$ (g/g)	1.18	11.0	1.42	13.0
$Y_{glu}$ (g/g)	1.73	23.0	1.25	18.0

Table 5. RMSE results for the PINN-PNSB model, updated physical model, PINN-derived model, and classical ANN on training/identification datasets.

RMSE	x	Exp 1-5	
		fru	glu
PINN	0.0459	0.0155	0.0538
Updated model	0.141	0.0339	0.166
PINN-derived model	0.153	0.0284	0.163
Classical ANN	0.0787	0.0337	0.0882

Table 6. RMSE results for the PINN-PNSB model, updated physical model, PINN-derived model, and classical ANN on the test dataset.

RMSE	x	Exp 6	
		fru	glu
PINN	0.167	0.0745	0.234
Updated model	0.404	0.145	0.450
PINN-derived model	0.212	0.0131	0.247
Classical ANN	0.362	0.0951	0.344

## 5. CONCLUSION

A PINN-based hybrid model is designed to improve the predictive performance of the mathematical model for PNSB cultures. An existing physical model is well-adjusted to experimental data but suffers a performance deterioration once new conditions are incorporated. To handle this problem, a PINN-PNSB model is designed, combining physical knowledge from the initial model with experimental results. The shallow PINN architecture was selected by comparing the final losses and training durations from pre-selected candidates containing a distinct number of neurons. A two-stage training is preferred for the final training, considering that it leads to lower residual losses than the single-stage one. PINN training is completed in a few minutes, and in parallel, the model parameters are re-identified, producing a PINN-derived model. Following training, the performance of the PINN-PNSB model is compared to a revised mechanistic model, a classical ANN, and the physical model whose parameters were optimized during the PINN training. RMSE results from the training phase indicate that the PINN-based model delivers the best performance, followed by the pure ANN and the PINN-derived model. Concerning parameter uncertainty, the updated physical model compensates for having the worst predictive accuracy by exhibiting the best results among the parametric models. Finally, when tested on unseen data, the PINN-PNSB model again outperforms its counterparts, with the PINN-derived model ranking second. These findings confirm the suitability of PINN-based models in improving predictive accuracy in bioprocesses, especially in small data scenarios. Future work should focus on expanding the application of PINN-based models to other PNSB cultures where more complex physical structures and additional inputs are present. Further analysis is also needed to investigate alter-

native architectures, particularly those specifically designed for time-series data.

## ACKNOWLEDGEMENTS

The research leading to these results has been funded by the Public Service of Wallonia (Economy, Employment and Research), under the FoodWal agreement n°2210182 from the Win4Excellence project of the Wallonia Recovery Plan.

## REFERENCES

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, 265–283.
- Bangi, M.S.F., Kao, K., and Kwon, J.S.I. (2022). Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for  $\beta$ -carotene production using *saccharomyces cerevisiae*. *Chemical Engineering Research and Design*, 179, 415–423.
- Cui, T., Bertalan, T., Ndauro, N., Khare, P., Betenbaugh, M., Maranas, C., and Kevrekidis, I.G. (2024). Data-driven and physics informed modeling of chinese hamster ovary cell bioreactors. *Computers & Chemical Engineering*, 183, 108594.
- Fekih-Salem, R., Dewasme, L., Castro, C.C., Nobre, C., Hantson, A.L., and Vande Wouwer, A. (2019). Sensitivity analysis and reduction of a dynamic model of a bioproduction of fructo-oligosaccharides. *Bioprocess and Biosystems Engineering*, 42, 1793–1808.
- Hülse, T., Barnes, A.C., Batstone, D.J., and Capson-Tojo, G. (2022). Creating value from purple phototrophic bacteria via single-cell protein production. *Current Opinion in Biotechnology*, 76, 102726.
- Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.
- Lagergren, J.H., Nardini, J.T., Baker, R.E., Simpson, M.J., and Flores, K.B. (2020). Biologically-informed neural networks guide mechanistic modeling from sparse experimental data. *PLoS computational biology*, 16(12), e1008462.
- Lu, L., Meng, X., Mao, Z., and Karniadakis, G.E. (2021). Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1), 208–228.
- Nunes, M.C., Dewasme, L., Gilson, M., Bayon-Vicente, G., Leroy, B., and Vande Wouwer, A. (2024). Robust tube-based predictive control of continuous protein production by purple non-sulfur bacteria. *IFAC-PapersOnLine*, 58(14), 724–729.
- Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378, 686–707.
- Rogers, A.W., Song, Z., Ramon, F.V., Jing, K., and Zhang, D. (2023a). Investigating ‘greyness’ of hybrid model for bioprocess predictive modelling. *Biochemical Engineering Journal*, 190, 108761.
- Rogers, A.W., Cardenas, I.O.S., Del Rio-Chanona, E.A., and Zhang, D. (2023b). Investigating physics-informed neural networks for bioprocess hybrid model construction. In *Computer Aided Chemical Engineering*, volume 52, 83–88. Elsevier.