12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Model Predictive Control using Physics Informed Neural Networks for Process Systems.

Rahul Patel, Sharad Bhartiya, Ravindra Gudi*

Department of Chemical Engineering, Indian Institute of Technology Bombay, Mumbai 400076, India

**(E-mail: ravigudi@iitb.ac.in)*

Abstract: Model based control approaches require an accurate and computationally fast prediction model to solve the governing equations in real time. While numerical approaches based on first principles models are accurate, the high computational cost renders them unsuitable for online estimation and model predictive control (MPC). On the other hand, the reduced order models can provide real time solutions, but there is invariably a trade-off between the accuracy and computational time. Machine learning based approaches such as physics informed neural networks (PINN) that are based on incorporating physics-based knowledge into NNs, can provide faster and accurate solutions in such scenarios. This work demonstrates the control-oriented modeling of process systems governed by ODEs and DAEs using residual PINN that is trained using neural tangent kernel update. Such PINN models can replace the conventional numerical time integration of the process dynamics and facilitate accurate and faster predictions. We present our results for setpoint tracking via MPC using the PINN model to demonstrate the capability of the proposed approach.

Keywords: Model Predictive Control, Physics Informed Neural Networks, Machine Learning, Optimization, State Estimation.

1. INTRODUCTION

Model-based approaches for the control of chemical processes require accurate dynamic process models. Chemical process systems are generally nonlinear and complex in nature. First principles modeling these dynamic process systems results in a set of ordinary differential equations (ODE), partial differential equations (PDE), or differential algebraic equations (DAE). Analytical solutions for such systems are usually unavailable, and one has to rely on numerical methods to obtain accurate solutions. These numerical methods are typically discretization based and have high computational costs. However, process models need to be solved faster and in real time for applications in estimation and control. Hence, using numerical methods to solve differential equations (DE) becomes challenging for real processes.

Another aspect is that first principles dynamic models are expensive to obtain and may require significant data to obtain parameters for accurate predictions. The alternative of obtaining data driven dynamic models is therefore widely practiced. For real time control, even though reduced order modeling (ROM) approaches exist, there is invariably a trade-off between model complexity, accuracy, and computational time. Machine Learning (ML) algorithms such as Neural Networks (NN) can represent the knowledge inferred from these process systems in the form of nonlinear mappings that can overcome the issue of high computational cost. However, these approaches are primarily data-driven and are biased to represent the information available from the training data. NNs can suffer from poor generalization on unseen data in case of a small amount of training data that could also be corrupted with noise.

Since model-based approaches, such as Model Predictive Control (MPC), require the repetitive and recursive solution of the underlying nonlinear DEs in real time, use of an accurate, robust, and computationally inexpensive model becomes inevitable in such cases. NNs can be regularized by the available knowledge from the physics of the system to provide an accurate and fast prediction model. Recently, Raissi et al. (2019) introduced Physics Informed Neural Networks (PINN) to train the NNs by using the governing equations of the physical system. PINNs incorporate the governing DEs in the loss function to train the NNs. The quantity of data required to train the NN model can be reduced using this physics-based regularization while ensuring that the trained model is constrained to satisfy the underlying physical laws. PINNs have been shown to solve the DEs accurately and have found applications in areas related to high-speed aerodynamic flows, nano-optics and metamaterials, and heat transfer problems. PINNs rely on automatic differentiation (AD) to compute the derivative, hence providing an efficient and accurate alternative to conventional approaches (Baydin et al., 2018).

Conventionally, PINNs are trained on the spatial variables and time as inputs to the NN to solve DEs, and does not require solution data to be trained. These NNs are trained on a fixed values of manipulated variables (MV), initial conditions (IC), other process parameters, and a specified time horizon and cannot predict outputs for varying MVs and longer time horizons as encountered in process control systems. Antonelo et al. (2021) proposed a PINN-based approach to simulate for longer time horizons. The MVs and ICs are fed as inputs to the NN, and the trained model is used recursively to predict over a longer time horizon. In our earlier work, we used these physics augmented NNs for state estimation and demonstrated the advantages of PINNs for cases with a) noisy plant measurements and b) corrupted plant models (Patel et al., 2022a, b). Nicodemus et al. (2022) demonstrated MPC on a multi-ink manipulator governed by first order ODE solved using PINN. Similarly, (Zheng and Wu, 2023) demonstrated MPC using physics informed recurrent neural network, but the derivatives of the states were not obtained by AD. In another work, the ability of PINNs to solve PDEs and use the closed analytical form for temperature trajectory optimization of a tubular reactor was explored (Patel et al., 2023). These studies show the effectiveness of PINNs for representative problems in systems engineering.

PINNs usually require many layers for better training and can suffer from convergence issues due to the nature of the multiple nonlinear constraints resulting from representing the physics related equations. To overcome this, we propose to use the Residual Neural Networks (ResNet) for PINNs for improved training, coupled with a Neural tangent kernel (NTK) update based gradient descent to address the convergence issues (Wang et al., 2022). We propose to use this high-fidelity PINN model as the prediction model instead of ROMs for online optimization in MPC. Such PINN based model can replace the conventional numerical time-integration of the nonlinear dynamics while being computationally faster and retaining sufficient accuracy. While there have been a few applications of conventional PINNs to model ODE and PDEs, their use to model stiff systems such as a DAE system and also to demonstrate their use for MPC applications has not been fully explored in literature.

In this paper, we evaluate the proposed Residual-PINN based control modelling approach on a representative system of Continuous Stirred Tank Reactor (CSTR) governed by nonlinear ODEs and an electrochemical system governed by DAE and demonstrate MPC. In this regard, the highlights of this work are as follows:

1. Dynamic modeling of nonlinear ODE and DAE system using residual PINN with an NTK update strategy.
2. Model predictive control of a CSTR represented by a system of ODEs.
3. Model predictive control of an electrochemical system represented by DAEs.

The remainder of the paper is structured as follows: Section 2 outlines the methodology applied. Section 3 discusses the representative ODE and DAE systems to be modeled by PINN. Section 4 discusses PINN modeling and MPC results, followed by summarizing conclusions in Section 5. *The supplementary information (SI) containing the Tables of model parameters and MPC parameters is available at:* *https://github.com/SCEPTRE-Lab/ADCHEM/*

2. METHODOLOGY

This section briefly introduces the NN and its ResNet variant to represent the physics based governing equations and generate a PINN model. Further, we present short preliminaries on MPC that is based on the PINN model for model predictions. We refer the reader to Bishop and Nasrabadi (2006) and Rawlings et al. (2017) for more details on NN and MPC.

2.1 PINN preliminaries

Consider a system governed by DAEs, collectively represented by function F given by (1) where the independent variable is time t and the solution of the DAE is Y (consisting of differential and algebraic states). Here U is exogenous input, also referred to as the MV of the system.

$$F\left(\frac{dY}{dt}, Y, U, t, \ldots\right) = 0 \qquad (1)$$

The initial conditions of the states for the DAEs are given as:

$$Y(0) = Y_0 \qquad (2)$$

In the proposed PINN approach, the solution of any system of DAEs is obtained by considering a trial solution $\bar{Y}$, given by the NN as in (3).

$$\bar{Y}(X) = NN(X, \theta) \qquad (3)$$

Here $NN(X,\theta)$ represents a feedforward NN with input X (having initial states, U, and t) and θ are the parameters of the NN. For deep NNs, providing a skip connection to the hidden layers improves the training. Such an NN that stacks residual blocks on top of each other to form a network is called a ResNet $\mathcal{N}(X,\theta)$, defined by (4)

$$\bar{Y}(X) = \mathcal{N}(X,\theta) = NN(X,\theta) + X \qquad (4)$$

The parameters of the NN are obtained by minimizing the loss function to accurately capture the relationship $F\colon X \longrightarrow Y$. For a DAE system represented by the general form as given by (1), the loss function to train the PINN is formed by considering this as the trial function, as shown in (5).

$$\mathcal{L}_{PINN}(X,\theta) = \sum_{j=1}^{D} \left\| F\left(\frac{d\bar{Y}_j}{dt}, \bar{Y}_j, U, t\right) \right\|^2 + \sum_{i=1}^{C} \|\bar{Y}_{i0} - Y_{i0}\|^2 \qquad (5)$$

The first term in (5) captures the residuals from the DAE evaluated at the internal collocation points for inputs X, and second term ensures that the initial conditions are satisfied. D and C are the number of internal collocation points and initial points for $t = 0$, respectively. The same approach to formulate NN and its loss function applies to an ODE system. Such a loss term penalizes the deviation of predicted values if they violate the governing equation by evaluating the residual and gives better predictions.

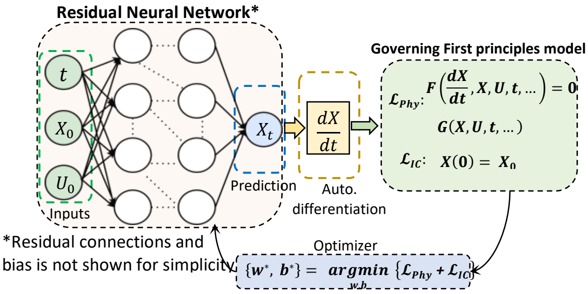
Residual Neural Network*
t
X_0
U_0
Inputs
X_t
Prediction
$\frac{dX}{dt}$
Auto. differentiation
Governing First principles model
$\mathcal{L}_{Phy}$: $F\left(\frac{dX}{dt}, X, U, t, \ldots\right) = 0$
$G(X, U, t, \ldots)$
$\mathcal{L}_{IC}$: $X(0) = X_0$
*Residual connections and bias is not shown for simplicity
Optimizer
$\{w^*, b^*\} = \underset{w,b}{argmin}\, \{\mathcal{L}_{Phy} + \mathcal{L}_{IC}\}$

Figure 1. Schematic representation of the PINN training framework

Fig. 1 shows the PINN training framework where an NN predicts the outputs whose derivatives are obtained using AD to further evaluate the residuals that form the loss function. Backpropagation algorithms, such as the gradient descent method, are widely used to optimize the loss function. The loss terms can be weighed suitably to give more importance to a specific output and the initial condition. However, this optimization problem to obtain the PINN parameters could be non-convex due to the nonlinear constraints resulting from representing the governing equations. To address these convergence issues, Wang et al. (2022) proposed a novel gradient descent using the NTK.

$$\mathcal{K}(Y,Y';\theta) = \nabla_\theta \mathcal{N}(Y;\theta) \cdot \nabla_\theta \mathcal{N}(Y';\theta) \tag{6}$$

For an NN, the NTK denoted by $\mathcal{K}$ can be written as a dot product between inputs with the gradient of the NN function as shown in (6). The PINN training algorithm utilizes eigenvalues of the NTK to adaptively calibrate the convergence rate of the total training error.

2.2 Basics of Model Predictive Control

MPC refers to a class of control approaches developed initially in the process industry for applications such as disturbance rejection, reference tracking, or general cost function optimization. They are widely popular due to their capability to handle operating constraints and multi-variable interactions.

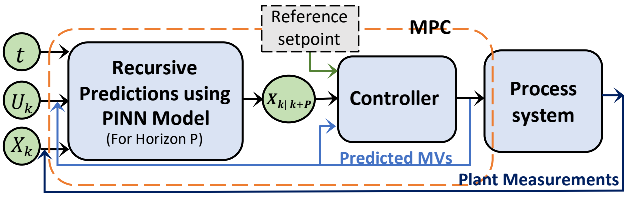

Figure 2. Representative diagram for MPC using the PINN model.

Any MPC formulation requires a prediction model that predicts the future behaviour over a horizon P at each sampling instant. For setpoint tracking application, the MPC problem at the sampling instant k, can be defined as a constrained optimization problem, whereby the future M manipulated inputs are obtained by minimizing a quadratic objective function defined as follows:

$$J = \alpha * \sum_{j=1}^{P} \left\| X_{Rk+j} - \hat{X}_{k+j} \right\|^2 + \beta * \sum_{i=0}^{M-1} \|\Delta U_{k+i}\|^2 \tag{7}$$

The optimization problem becomes,

$$U_{k,opt} = \underset{U}{argmin} \left\{ J(X_{Rk+j}, \hat{X}_{k+j}, \Delta U_{k+i} \right\}$$

While being subject to,

$$\hat{X}_{k+j} = \mathcal{N}\left(X_{k+j-1}, U_{k+j-1}, k\right)$$

$$\Delta U_{k+i} = U_{k+i} - U_{k+i-1} \leq \Delta U_{Max}$$

$$U_{low} \leq U_{k+i} \leq U_{high}$$

X_{Rk+j} is the reference setpoint *(j=1, 2,...P)* for horizon P, $\hat{X}_{k+j}$ is the prediction (by solving nonlinear DEs or by PINN), and U_{k+i} is the manipulated input *(i=1, 2,...M)* for horizon M at k^{th} instant. α and β are matrices to suitably weigh individual states and penalize sudden large changes in the manipulated inputs, respectively. The MV and the predictions are bounded by the range $[U_{low}, U_{high}]$. The maximum move that a MV can make is constrained by ΔU_{Max}.

The PINN-based model is used recursively to predict the future for any given time horizon. As illustrated in Fig. 2, these predictions are fed to a controller that performs online optimization to generate the sequence of MVs to track the reference setpoint. The plant measurements obtained by implementing the first MV move is fed back to the PINN to give recursive predictions.

3. REPRESENTATIVE CASE STUDIES

We discuss representative process systems to evaluate the proposed control oriented modeling in this section.

3.1 Case Study A: Non-isothermal CSTR represented by a system of ODEs

A non-isothermal CSTR governed by a system of nonlinear ODEs has been considered to demonstrate the control-oriented modeling using PINN (Marlin, 1995). The dynamic model for a CSTR undergoing an irreversible first-order reaction:

$$A \longrightarrow B$$

is given by (8) and (9) as follows:

$$\frac{dX_1}{dt} = \frac{U_2}{V}(C_{A0} - X_1) - k_0 X_1 \, exp\left(-\frac{E}{RX_2}\right) \tag{8}$$

$$\frac{dX_2}{dt} = \frac{U_2}{V}(T_0 - X_2) + \frac{(-\Delta H_r)k_0}{\rho C_p} X_1 \, exp\left(-\frac{E}{RX_2}\right) - \frac{Q}{V\rho C_p} \tag{9}$$

$$\text{Where, } Q = \frac{a(U_1)^{b+1}}{U_1 + \left(\frac{a(U_1)^b}{2\rho_c C_{pc}}\right)}(X_2 - Tc_{in})$$

X_1 and X_2 are the reactor concentration of species A and the reactor temperature. U_1 and U_2 are the coolant flow rate and feed flow rate. The dynamic characteristics of the system depend on the set of parameter values given in Table S1.

Equations (8) and (9) represent the ground truth of the system. The plant model is assumed to be same as the ground truth for this study. For demonstrating the control-oriented modeling of the ODE system, the states $X = [X_1, X_2]$ are assumed to be fully observable and input MV are $U = [U_1, U_2]$. The plant measurements are obtained by corrupting both the differential states with process noise of mean 0 and standard deviation [2.64e-4, 3.93-1] (i.e., 0.1% of nominal steady-state values (SS)). The prediction model is obtained by training the PINN model on the ground truth.

3.2 Case Study B: An Electrochemical system represented by DAEs

An electrochemical system to represent the galvanostatic charge/discharge process of a thin film nickel hydroxide electrode can be modeled by DAEs (Çelik et al., 2002). The species balance equation gives a differential equation that shows the rate of change of the mole fraction of nickel hydroxide given by (10). The charge balance equation gives an algebraic equation (11) where j_1 and j_2 are derived using the

Butler–Volmer kinetics. Depending on the value of i_{app} the operation can be charging, open-circuit, or discharging.

$$\frac{\rho V}{W}\frac{dy_1}{dt} = \frac{j_1}{F} \tag{10}$$

$$j_1 + j_2 - i_{app} = 0 \tag{11}$$

$$j_1 = i_{01}\left[2(1-y_1)\exp\left(\frac{0.5F}{RT}(y_2-\phi_{eq,1})\right)-2y_1 \times \exp\left(\frac{-0.5F}{RT}(y_2-\phi_{eq,1})\right)\right] \tag{12}$$

$$j_2 = i_{02}\left[\exp\left(\frac{F}{RT}(y_2-\phi_{eq,2})\right) - \exp\left(\frac{-F}{RT}(y_2-\phi_{eq,2})\right)\right] \tag{13}$$

Here, the differential state y_1 is the mole fraction of nickel hydroxide and the algebraic state y_2 is the potential difference at the solid-liquid interface. The parameter values used in the model are available in Table S2.

Equations (10) – (13) represent the ground truth of the DAE system. Further, to demonstrate the control-oriented modeling of the DAE system, the full observable states are $X = [y_1 , y_2]$ and temperature (T) is considered as the MV (U). The plant measurements are obtained by corrupting both the differential and algebraic states with process noise of mean 0 and standard deviation [9.06e-4, 4.77e-4] (i.e., 0.1% of SS values).

The capability of the trained PINN model to represent the nonlinear dynamics of the process system is evaluated using these ODE and DAE systems. For this purpose, we develop a control-oriented PINN model that takes the current state and MV as input and predicts the future state at any given timestep. Such a control-oriented model is evaluated to estimate the states of the system for changes in MV and ICs. The following section discusses the results of this control-oriented modeling for estimation and control.

4. RESULTS AND DISCUSSION

The PINN model for the ODE system is trained on a dataset of 12800 samples with an NN of 4 layers with 64 neurons. Similarly, PINN to model the DAE system is trained on a dataset of 10000 samples with an NN of 4 layers with 64 neurons. Dataset refers to the randomly generated input collocation points. Hyperbolic tan was used as an activation function in both cases. The PINN was implemented and trained using Adam optimizer from Python's SciANN package (Haghighat & Juanes, 2021). The details on the input dataset to train the NN can be obtained in SI.

4.1 Case Study A: Non-isothermal CSTR represented by a system of ODEs

The multi-input multi-output (MIMO) CSTR system is modeled using PINN to predict future states and study the effects of MV. The PINN was trained on a timestep of 0.6 minutes and used recursively to predict the states for a longer time horizon. The initial states X_k of the system at every instant k are obtained from the plant model, i.e. ground truth corrupted with noise (refer to Section 3). Fig. 3a and 3b shows the recursive PINN estimates for 90 timesteps subject to the changes in MV given by Fig 3c and 3d. The actual states (shown by the violet line) are obtained by solving the nonlinear ODE model numerically using the LSODA algorithm and compared with the predictions obtained from the trained PINN model (dashed red line).

The results illustrated in Fig. 3 demonstrate that the PINN model can predict for longer time horizons and capture the effect of changes in both MVs. The bounds of MV on which the PINN model is trained are shown by dashed green lines. The region bounded by the dashed orange lines is when the MVs are beyond the range on which PINN was trained. It is seen that the PINN model performs poorly for extrapolation, and its predictions deviate from the actual states.

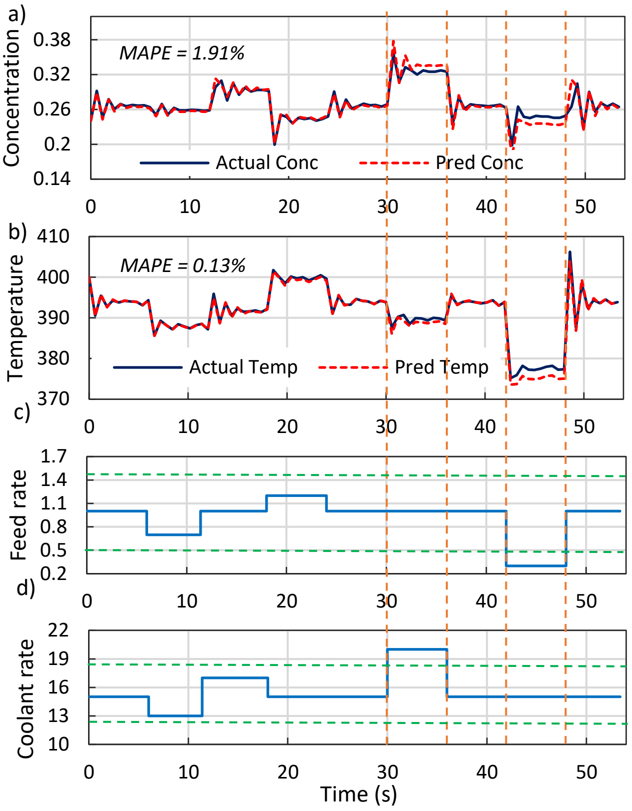

Figure 3. Recursive PINN predictions for varying MV for CSTR

The PINN-based prediction model is used in an online optimizer to demonstrate the MPC of the CSTR reactor to track the setpoint of reactor concentration. The prediction horizon (P) for the MPC is 20, and the control horizon (M) is 6. The MVs are constrained by the bounds given in Table S3 to limit their values in a given range. Also, the maximum change in MV that can occur in a timestep is limited to 1.2 and 0.2 for U_1 and U_2 respectively. Fig. 4a and 4b compare the results for the MPC using the PINN model as the prediction model with the MPC using the nonlinear ODE model. Fig. 4a shows that the PINN-based MPC can track the setpoint of the reactor concentration (shown by the dashed grey line), whereas Fig. 4b compares the predicted reactor temperature. The MVs predicted by both the MPC formulations are compared in Fig. 4c and 4d. Fig. 4 thus illustrates that the PINN model can be used in MPC applications to track setpoint changes and predict future control action.

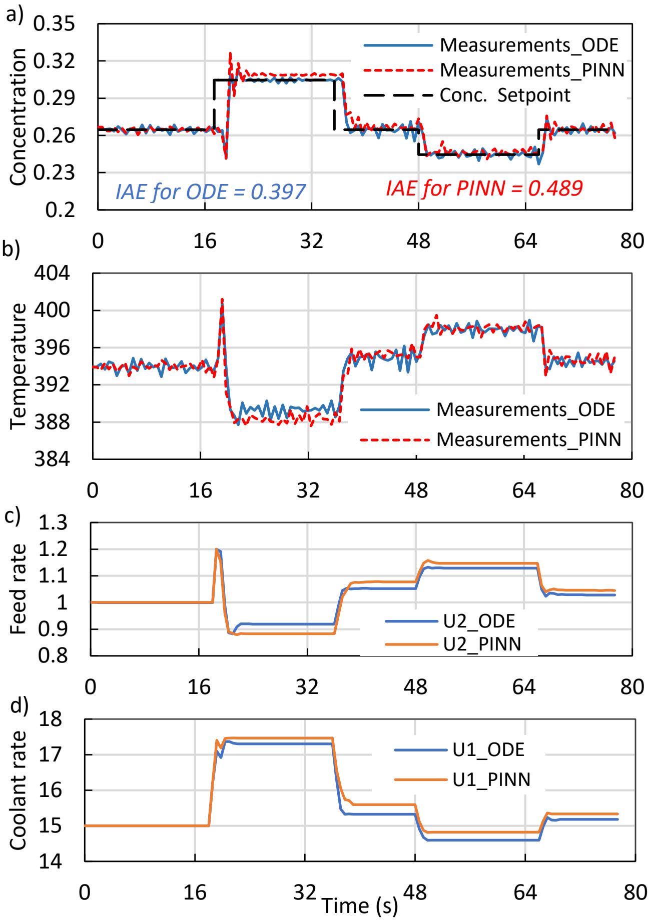

Figure 4. MPC results to track the setpoint of reactor concentration (PINN model vs Nonlinear ODE model)

4.2 Case Study B: An Electrochemical system represented by DAEs

The nonlinear DAE system is a single-input multi-output (SIMO) system with temperature as the manipulated variable. The solution of DAEs suffers from inconsistent initialization, as discussed in Çelik et al. (2002). The PINN model was trained on consistent initial values randomly generated using MATLAB's *decic* function for a timestep of 500s. Fig. 5 demonstrates the capability of the PINN model to predict the solution of DAE for different sets of ICs and temperatures. The solution of the DAE model is obtained from Gekko's IPOPT Solver, and it matches the PINN predictions.

A perturbation of magnitude ± 20 is applied to generate the step input signal to study the effect of changes in MV. Fig. 6 shows the states predicted by the PINN model, PINN in self-loop and the DAE model for the changes in MV. Self-loop refers to the model when the PINN predictions are fed back again as an input rather than using the actual plant measurements at instant k. The states predicted for a longer time horizon by the PINN model are close to those obtained from the DAE model.

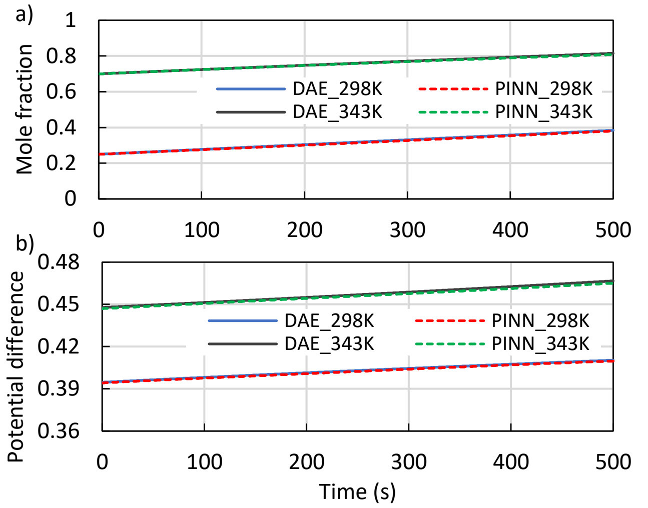

Figure 5. Comparison of PINN prediction at different initial conditions and manipulated inputs

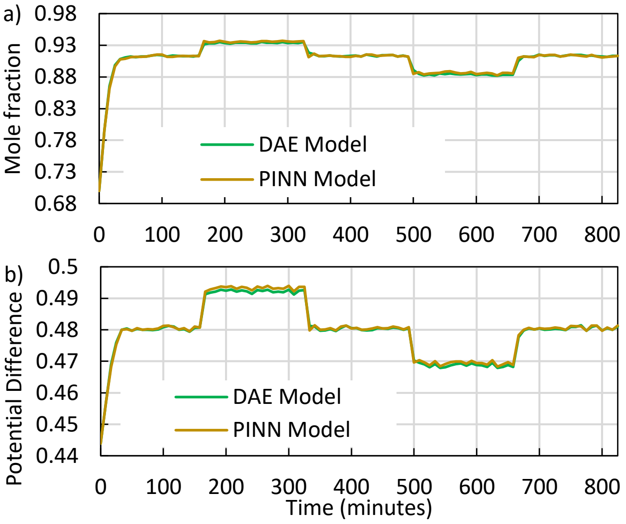

Figure 6. Recursive PINN prediction for step changes in MV

Similar to the ODE system, the PINN-based prediction model is used to demonstrate the MPC of the DAE system to track the setpoint of the mole fraction. The prediction horizon (P) for this MPC is 7, and the control horizon (M) is 3. The temperature (MV) is constrained to lie in the range of [273, 343]. Also, the maximum change in MV that can occur in a timestep is limited to 10. Fig. 7a and 7b compare the results for the MPC using the PINN model as the prediction model with the MPC using the Nonlinear DAE model. Fig. 7a shows that the PINN-based MPC can track the setpoint of mole fraction, whereas Fig. 7b compares the predicted potential difference. The MV predicted by both the MPC formulations are compared in Fig. 7c. As observed in Fig. 7, the PINN model can be used in MPC applications to track setpoint changes and predict future control action for the representative DAE system.

The average computational time taken by the MPC at each time step has been compared in Fig. 8 for two different MPC configurations of the DAE system. The computational time taken is significantly less than the prediction timestep. This also shows that the PINN model can be efficiently used in an online optimizer with smaller timesteps. Thus, the PINN-based prediction model can replace the time-consuming step of numerical integration and facilitate faster state estimation for both the ODE and DAE system.

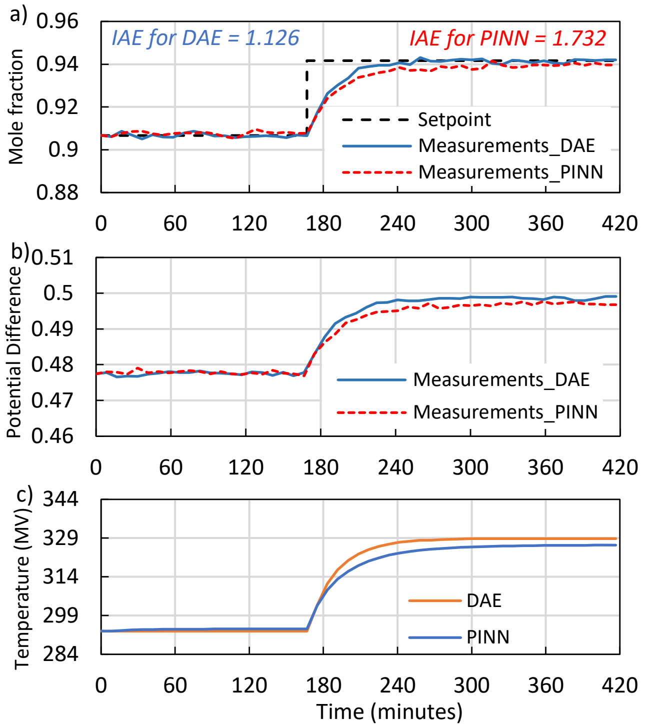

Figure 7. MPC results to track the setpoint of y_1 (PINN model vs DAE model)

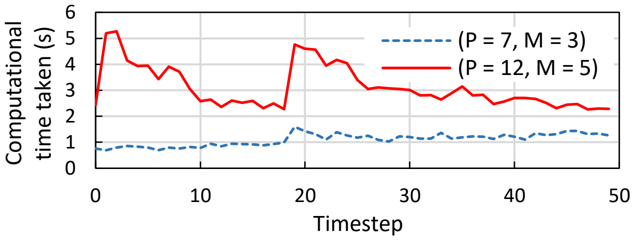

Figure 8. Computational time taken at each timestep by the MPC algorithm for DAE system.

5. CONCLUSION

PINN is a promising approach for modeling the dynamics of complex process systems by using the governing physics. The applicability of PINN to develop a control-oriented model was explored in this work. We evaluated the approach on a CSTR governed by a system of ODEs and an electrochemical system governed by DAEs for setpoint tracking using MPC. The presented results show that the PINN model can be used as a prediction model in MPC formulation since it facilitates faster prediction by replacing the time-consuming numerical integration. As the PINN model is an approximation, the nonlinear mapping obtained will not be as accurate as the conventional offline numerical methods to solve DEs. However, such PINN based models can offer a better alternative in scenarios where we have insufficient historical data to train conventional models or when the numerical approaches to solve DEs are computationally expensive to be used in real time. The limited extrapolation ability of the PINNs needs to be addressed, and formal ways to incorporate the structure of the governing DEs should be explored in further studies. More relevant cases of model plant mismatch and online parameter update are a matter of future work.

REFERENCES

Antonelo, E.A., Camponogara, E., Seman, L.O., Jordanou, J.P., de Souza, E.R. and Hübner, J.F., 2024. Physics-informed neural nets for control of dynamical systems. *Neurocomputing*, p.127419.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning, 18(153), pp.1-43*

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, Issue 4). Springer.

Çelik, E., Karaduman, E., & Bayram, M. (2002). Numerical method to solve chemical differential-algebraic equations. *International Journal of Quantum Chemistry*, *89*(5), 447–451.

Haghighat, E., & Juanes, R. (2021). SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, *373*, 113552.

Marlin, T. E. (1995). *Process control: designing processes and control systems for dynamic performance.* McGraw-Hill Education

Nicodemus, J., Kneifl, J., Fehr, J., & Unger, B. (2022). Physics-informed Neural Networks-based Model Predictive Control for Multi-link Manipulators. *IFAC-PapersOnLine*, *55*(20), 331–336.

Patel, R., Bhartiya, S., & Gudi, R. (2023). Optimal temperature trajectory for tubular reactor using physics informed neural networks. *Journal of Process Control*, *128*, 103003.

Patel, R., Bhartiya, S., & Gudi, R. D. (2022). State Estimation Using Physics Constrained Neural Networks. *2022 IEEE International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, 61–66.

Patel, R. S., Bhartiya, S., & Gudi, R. D. (2022). Physics Constrained Learning in Neural Network based Modeling. *IFAC-PapersOnLine*, *55*(7), 79–85.

Raissi, M., Perdikaris, P., & 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.

Rawlings, J. B., Mayne, D. Q., & Diehl, M. (2017). *Model predictive control: theory, computation, and design* (Vol. 2). Nob Hill Publishing Madison, WI.

Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, *449*, 110768.

Zheng, Y., & Wu, Z. (2023). Physics-Informed Online Machine Learning and Predictive Control of Nonlinear Processes with Parameter Uncertainty. *Industrial & Engineering Chemistry Research, 62*(6), 2804–2818.