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

Kinetics Parameter Identification of Chain Shuttling Polymerization Based on Physics-Informed Neural Networks

Jieming Zhao*, Zhou Tian*, Xixiang Zhang*,
Zhaoyang Duan*, Jingyi Lu*

** Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology
Shanghai, China, (e-mail: tianzhou@ecust.edu.cn)}*

Abstract: Chain-shuttling polymerization is widely used to synthesize specialized polymer materials with customized properties. The significance of modeling in chemical process simulation lies in accurately describing and analyzing complex chemical systems through mathematical and computational models, thereby enhancing the efficiency and reliability of process design. Due to limited and noisy measurements and the complex model structure, parameter identification for chain-shuttling polymerization has been a long-standing problem. To address this issue, in this work, we propose to first describe the dynamic process with a set of ordinary differential equations (ODEs) based on the method of moments. This method characterizes the dynamic variations of the average chain length. After that, we introduce Physics-Informed Neural Networks (PINNs) to estimate the unknown parameters in the ODEs. Since PINNs can incorporate the ODEs constraints during the training process, they can effectively integrate the polymerization mechanism with the observed process data, thereby reducing the amount of data needed for training. A comparative analysis between parameters estimated using PINNs and the ground truth values demonstrates high accuracy and efficiency, even with sparse and limited observations. This showcases the potential value of PINNs in chemical process identification.

Keywords: Polymerization process, Parameter identification, Method of moments, Chain-shuttling polymerization, Parameters identification, Physics-informed neural networks

1. INTRODUCTION

Polyolefins, a crucial class of high-polymer materials in chemical engineering, play an indispensable role in various fields such as plastics, rubber, fibers, and more. Chain shuttling polymerization is a distinctive process wherein monomer molecules can be transferred between different polymer chains during the reaction. This type of polymerization leads to the formation of branched polymers, thereby influencing the molecular weight distribution and structure of the resulting polymer and subsequently impacting its properties. The significance of chain-shuttling polymerization lies in its ability to control the structure and properties of polymers within the realm of polymer chemistry. However, multiple factors including reaction conditions and catalyst performance often influence the dynamic characteristics of polyolefin polymerization. Consequently, developing accurate kinetic models for simulating and predicting these processes has emerged as a key challenge in research.

In the pursuit of a deeper comprehension and effective control of polyolefin polymerization processes, parameter identification has garnered significant attention as a research field of paramount importance. The primary objective of parameter identification is to develop a robust methodology that allows for the precise estimation of the parameters governing the polymerization process through the analysis of experimental data and models. This, in turn, serves to enhance the accuracy and predictive capabilities of kinetic models. Research endeavors in this domain are dedicated to overcoming the inherent complexities and diversities associated with polyolefin polymerization, ultimately providing more practical and reliable modeling approaches tailored to meet engineering application requirements.

Fundamental statistical parameter estimation methods include the least square method, Bayesian estimation, and et al. Bystritskaya (Bystritskaya, Pomerantsev et al. (1999)) employed Bayesian estimation to predict the aging of polymeric systems in situations where directly measuring the desired properties is impossible or challenging. Another type of parameter estimation relies on metaheuristic algorithms, such as genetic algorithms (GA), particle swarm optimization algorithms (PSO), and others. For example, Prata (Prata, Schwaab et al. (2010)) developed a nonlinear dynamic data reconciliation procedure (NDDR) based on the particle swarm optimization (PSO) method. The procedure was validated online in real-time using actual industrial data obtained from an industrial polypropylene reactor.

The optimization algorithms mentioned above play an indispensable role in parameter identification problems. They possess several advantages, such as the capability to find global optimal solutions within relatively simple parameter spaces and robustness in dealing with local optima. Moreover, these algorithms exhibit excellent performance when executed with appropriate initial values and precise data conditions. However, a significant limitation of these algorithms is their substantial demand for experimental data, which can be a constraining factor in practical applications due to the associated high costs and time consumption.

In recent years, deep learning algorithms have made significant strides in various domains, expanding their application scope. Raissi (Raissi, Perdikaris et al. (2019)) conducted research on solving differential equations, exploring machine learning approaches that can effectively integrate both data and physical information. They proposed Physics-Informed Neural Networks (PINNs), which integrate physical information into the learning process. This model is capable of addressing physical problems involving complex ordinary and partial differential equations, overcoming the limitations of traditional machine learning methods in terms of data-driven and black-box modeling. It enhances the robustness and computational accuracy of the model while improving the interpretability of models constrained by physical laws. In order to tackle the challenges posed by complex parameter spaces and noisy data in traditional algorithms, utilizing Physics-Informed Neural Networks (PINNs) to solve the inverse problem of parameter estimation is considered an effective method (Cuomo, Di Cola et al. (2022)).

PINNs combine neural networks with physical models to perform parameter estimation by learning physical laws ((Karniadakis, Kevrekidis et al. (2021))). This not only enhances the accuracy of parameter identification but also reduces data requirements. Another advantage of PINNs is their flexibility, as they can adapt to various problems and models, thus providing a more feasible and reliable solution for parameter identification in engineering fields (Chen, Lu et al. (2020)).

Numerous scholars have conducted research on the application of PINNs in solving inverse problems. Daneker Mitchell employed the PINNs method to model the ultradian endocrine system for glucose-insulin interaction and estimate parameters of biological systems (Daneker, Zhang et al. (2023)). In this paper, a new and simple-to-implement "systems-biology-informed" deep learning algorithm is presented. This algorithm can reliably and accurately infer the hidden dynamics described by a mathematical model in the form of a system of ordinary differential equations (ODEs). Additionally, Son (Ngo and Lim (2021)) developed a Physics-Informed Neural Networks (PINNs) model to address the isothermal fixed-bed (IFB) model of catalytic carbon dioxide methanation. They utilized a forward PINNs model to construct the IFB packed-bed reactor model and employed a reverse PINNs model to uncover unknown effective factors involved in reaction kinetics.

If you want to apply PINNs to identify the dynamic parameters of chain-shuttling processes, a key point is to "embed" the physical model of chain-shuttling polymerization into PINNs. Since PINNs typically incorporate the model's information through ODEs we can apply the method of moments to simulate chain-shuttling polymerization (Mastan and Zhu (2015)). In the polymerization process, the method of moments is a widely used deterministic modeling approach. It provides an overall description of the polymerization process, capturing the dynamic behavior and statistical characteristics of polymer chains. It also derives a system of ODEs to reveal the average properties of polymers, such as the weight-average molecular weight (Zhang, Karjala et al. (2010)). These differential equations typically involve multiple kinetic parameters, which may be challenging to directly measure or estimate. Traditional parameter identification methods may require a substantial amount of experimental data and numerical techniques to estimate these parameters. However, these methods can be limiting in complex scenarios. PINNs can integrate the system's differential equation model of the polymerization process and automatically identify the parameters within the differential equations using sparse, noisy data. PINNs achieve parameter estimation by integrating physical principles and existing data, eliminating the necessity for manual intervention or intricate numerical computations.

In this paper, we propose to identify the key kinetic parameters in chain shuttling polymerization process using PINNs. In particular, we establish a set of ordinary differential equations that describe the dynamic variations of the average chain length using the method of moment matching. After that, we conduct an identifiability test to check if the ODEs are structurally identifiable. For the processes that are structurally identifiable, we directly incorporate the unknown kinetic parameters into the PINNs model and estimate them based on optimization algorithms such as Adam. Otherwise, we sort the unknown parameters via sensitivity analysis and fix the insensitive ones with prior knowledge to ensure the identifiability of the ODEs. The remaining parts of the paper are organized as follows: Section 2 introduces the structure of the PINNs. Section 3 models the chain-shuttling polymerization process using the method of moments, embeds the model into the PINNs and conducts parameter identifiability analysis on the model. Section 4 mainly focuses on the experimental section, where the constructed PINNs model is utilized for identifying the kinetic parameters. Section 5 summarizes the main conclusions of this thesis and proposes recommendations for future work.

2. PHYSICS-INFORMED NEURAL NETWORKS MODEL

In this section, we will briefly discuss the fundamental structure of PINNs and explore how inverse PINNs can assist in solving parameter identification problems.

2.1 Deep neural networks

Deep neural networks (DNNs) form the foundation of PINNs. DNNs operate through recursive linear and nonlinear transformations of input data, essentially functioning as compositional functions. Various types of DNNs architectures have been developed, including convolutional neural networks and recurrent neural networks, but for the purpose of this discussion, we will focus solely on fully connected neural networks (FNNs). The inherent differentiability of FNNs facilitates their seamless integration into automatic differentiation frameworks, enabling the efficient computation of gradients during the training process. This capability plays a pivotal role in addressing a wide range of problems where adjusting network parameters is essential to approximate or align with real-world phenomena. Furthermore, automatic differentiation provides a convenient mechanism for handling complex gradient propagation, ensuring the efficient updating of parameters even in deep neural networks.

To define an FNNs, in the ℓ^{th} layer, we define a weight matrix w^{ℓ}, bias b^{ℓ},and an activation function σ. Each hidden layer recives and outputs $x^{k-1} \in R^{N_{k-1}}$ from the previous layer. The transformation between layers is expressed as

$$\mathcal{N}^{\ell}(\boldsymbol{x}) = \sigma\left(\boldsymbol{w}^{\ell}\mathcal{N}^{\ell-1}(\boldsymbol{x}) + \boldsymbol{b}^{\ell}\right), \ for\ 1 \le \ell \le L-1\,. \quad (1)$$

2.2 Physics-Informed Neural Networks

A general PINNs model primarily consists of a neural network (typically an FNNs) and a physical information model defined by differential equations. When solving forward or inverse problems, the configuration of the model may vary slightly.

We shall commence by focusing on the task of forward PINNs problem for partial differential equations.

$$u_t + \mathcal{N}[u] = 0, x \in \Omega, t \in [0, T]\,. \quad (2)$$

Here $u_t(t, x)$ denotes the solution of ODE, $\mathcal{N}[\cdot]$ is a nonlinear differential operator, which represents the various derivatives of u with respect to its different orders, excluding u_t and their mutual interactions. Moving all terms of the equation to the left-hand side yields the general form of the physical differential equation show as Eq (3). We use a deep neural network to approximating $u\,(t, x)$

$$f(u, t) := u_t + \mathcal{N}[u]\,. \quad (3)$$

The architecture of the forward PINNs problem is shown in Figure 2.1. The objective of the forward PINNs problem is to solve the provided governing equation with initial, boundary, and operating conditions.

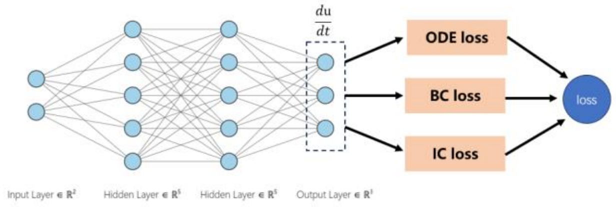


Fig2.1 Architecture of the physics-informed neural networks for generic forward problems

The input and output datasets of the FNNs were randomly sampled from the initial and boundary conditions provided by the governing equations during the training stage. The activation function σ, such as the sigmoid and hyperbolic tangent (tanh), was applied to each neuron.

In the forward problem, we define the loss function as follow:

$$Loss = MSE_u(\boldsymbol{\theta}) + MSE_{ode}(\boldsymbol{\theta})\,, \quad (4)$$

where

$$\mathrm{MSE_u} = \frac{1}{\mathrm{N_u}}\sum\nolimits_{\mathrm{i=1}}^{\mathrm{N_u}} |\ \mathrm{u}(\mathrm{t_u^i}, \mathrm{x_u^i}) - \mathrm{u^i}\ |^2\ , \quad (5)$$

and

$$MSE_{ode} = \frac{1}{N_{ode}}\sum\nolimits_{i=1}^{N_{ode}} |\ f\left(t_f^i, x_f^i\right)\ |^2\ . \quad (6)$$

$\mathrm{MSE_u}$ represents the mean squared error between the neural networks' fitted values and the ground truth values at initial and boundary conditions. MSE_{ode} represents the mean squared error between the neural networks and the actual physical laws. All the weights and biases are the parameters (θ) of the neural network. To optimize the neural networks.

$$\boldsymbol{\theta}^{*} = arg\ \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \quad (7)$$

The weights w^{ℓ},and biases b^{ℓ} for the ℓ^{th} hidden layer must be adjusted to minimize the loss function (Loss). The automatic differentiation (AD) process for computing spatial derivatives $(\frac{du}{dt})$ was performed using the reverse accumulation mode, which involves the retrograde propagation of derivatives starting from a specified output.

2.3 PINNs in the inverse problem

When presented with a model containing unknown parameters, PINNs integrate observational data and physical principles into the neural network constraints. This integration enables us to estimate the unknown parameters through PINN training. This involves the application of PINNs to solve the inverse problem. The parameters $\boldsymbol{\theta}$ of the FNNs are the optimized variables in the forward PINNs problem, while unknown model parameters are identified in the inverse PINNs using the optimized $\boldsymbol{\theta}^{*}$ obtained from the forward PINNs. Instead of using the initial condition as the training data, the inverse PINNs problem utilizes observation data from an external source, such as experimental data.

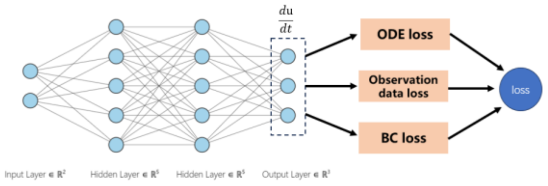


Fig2.2 Architecture of the PINN for generic inverse problems. (Inverse PINN takes observed data as constraints for training.)

In the inverse PINNs， the total loss is defined as follows:

$$Loss = MSE_{obs}(\boldsymbol{\theta}, \boldsymbol{p}) + MSE_{ode}(\boldsymbol{\theta}, \boldsymbol{p})\}, \quad (8)$$

which is a function of both neural networks' parameters θ and model parameters p. Here $\mathrm{MSE_{obs}}$ and MSE_{ode} are defined as follows:

$$\mathrm{MSE_{obs}} = \frac{1}{\mathrm{N_{obs}}}\sum_{\mathrm{i=1}}^{\mathrm{N_{obs}}} \mid \mathrm{u}(\mathrm{t_u^i}, \mathrm{x_u^i}) - \mathrm{u^i} \mid^2, \qquad (9)$$

$$MSE_{ode} = \frac{1}{N_{ode}}\sum_{i=1}^{N_{ode}} \mid f(t_f^i, x_f^i) \mid^2. \qquad (10)$$

All parameters can be identified by minimizing the loss function

$$\boldsymbol{\theta}^*, \boldsymbol{p}^* = arg\,\min_{\boldsymbol{\theta},\boldsymbol{p}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{p}), \qquad (11)$$

where $\mathrm{N_{obs}}$ is the number of observation data points (or the experimental data). $\mathrm{MSE_{obs}}$ is evaluated for the observation data. The MSE_{ode} enforces the structure imposed by the system of ODEs given in Eq (3).

3. PINNs-BASED CHAIN-SHUTTLING POLYMERIZATION AND PARAMETER ESTIMATION

In this section, we introduce the mechanism of CSA polymerization and the method of moments modeling based on the reaction mechanism. Finally, the application of inverse PINNs is introduced.

3.1 Ethylene chain shuttling polymerization mechanism

The process employed for ethylene chain shuttling polymerization comprises multiple distinct stages (Arriola, Carnahan et al. (2006)), Firstly, there is the catalyst initiation involving ethylene, where ethylene participates in initiating the polymerization reaction. Subsequently, there is chain propagation, which is the phase where polymer chains continuously grow as monomers are added to the active sites on the catalyst. Following this is the occurrence of chain transfer events, which facilitate the movement of the polymer chain from one active site to another. Catalyst deactivation is a stage where the catalyst undergoes deactivation, limiting its ability to further contribute to chain growth. The subsequent phase involves chain shuttling to a virgin Chain Shuttling Agent (CSA), providing an alternative pathway for extended polymerization. Finally, there is the phase of chain shuttling to polymerize-CSA, where a dormant polyethylene chain becomes bonded to a CSA molecule. Table 3.1 provides an overview of the polymerization phases.

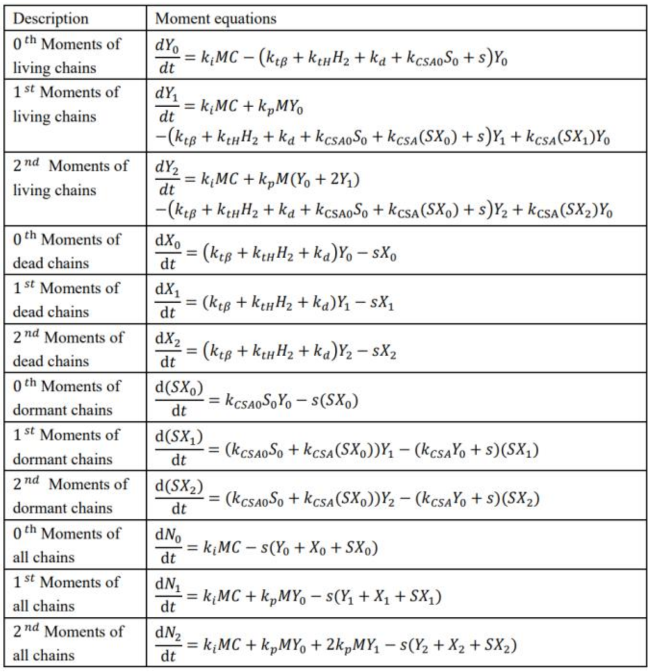
Table 3.1. Mechanism for ethylene chain shuttling polymerization.

Description	Chemical equations	Rate constants
Initiation	$C + M \rightarrow P_1$	k_i
Propagation	$P_r + M \rightarrow P_{r+1}$	k_p
β-hydride elimination	$P_r \rightarrow D_r + C$	$k_{t\beta}$
Chain transfer to hydrogen	$P_r + H_2 \rightarrow D_r + C$	k_{tH}
Deactivation of growing chain	$P_r + H_2 \rightarrow D_r + C$	k_d
Deactivation of active catalyst	$C \rightarrow C_d$	k_d
Chain shuttling to CSA	$P_r + S_0 \rightarrow SP_r + C$	k_{CSA0}
Chain shuttling to dormant chain	$P_r + SP_s \rightarrow P_s + SP_r$	k_{CSA}

3.2 Process modeling based on the method of moments

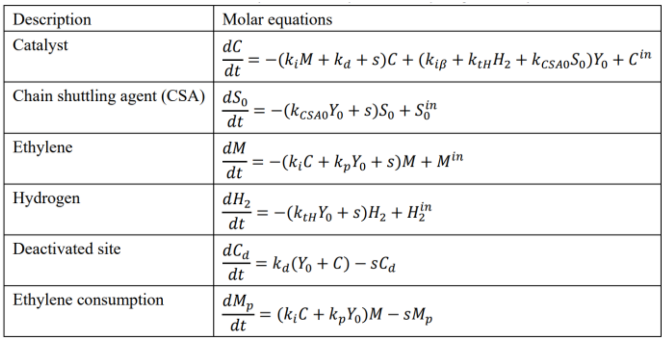
We need to establish a differential equation model that can describe the polymerization dynamics. The method of moments is a versatile deterministic approach widely employed in modeling various polymerization reaction processes. The principle involves partitioning the chain-length species within a polymerization reaction into finite intervals and then establishing mass balance equations with the average of each interval as an independent variable. These equations can be used to calculate the concentration and molecular weight distribution of different species in the polymerization reaction. Therefore, we developed a dynamic simulation model for ethylene chain-shuttling polymerization using a single-catalyst in a continuous stirred-tank reactor (CSTR) through the method of moments. In a CSTR, the flow rates entering and leaving the reactor need to be taken into account. The differential equations model obtained by the method of moments is shown in Table 3.2. Table 3.3 shows the material conservation equation.

Table 3.2 Moments equations in a dynamic CSTR for ethylene chain shuttling polymerization.



Description	Moment equations
0 th Moments of living chains	$\frac{dY_0}{dt} = k_iMC - (k_{t\beta} + k_{tH}H_2 + k_d + k_{CSA0}S_0 + s)Y_0$
1 st Moments of living chains	$\frac{dY_1}{dt} = k_iMC + k_pMY_0$ $-(k_{t\beta} + k_{tH}H_2 + k_d + k_{CSA0}S_0 + k_{CSA}(SX_0) + s)Y_1 + k_{CSA}(SX_1)Y_0$
2 nd Moments of living chains	$\frac{dY_2}{dt} = k_iMC + k_pM(Y_0 + 2Y_1)$ $-(k_{t\beta} + k_{tH}H_2 + k_d + k_{\mathrm{CSA0}}S_0 + k_{\mathrm{CSA}}(SX_0) + s)Y_2 + k_{\mathrm{CSA}}(SX_2)Y_0$
0 th Moments of dead chains	$\frac{\mathrm{d}X_0}{\mathrm{d}t} = (k_{t\beta} + k_{tH}H_2 + k_d)Y_0 - sX_0$
1 st Moments of dead chains	$\frac{\mathrm{d}X_1}{\mathrm{d}t} = (k_{t\beta} + k_{tH}H_2 + k_d)Y_1 - sX_1$
2 nd Moments of dead chains	$\frac{\mathrm{d}X_2}{\mathrm{d}t} = (k_{t\beta} + k_{tH}H_2 + k_d)Y_2 - sX_2$
0 th Moments of dormant chains	$\frac{\mathrm{d}(SX_0)}{\mathrm{d}t} = k_{CSA0}S_0Y_0 - s(SX_0)$
1 st Moments of dormant chains	$\frac{\mathrm{d}(SX_1)}{\mathrm{d}t} = (k_{CSA0}S_0 + k_{CSA}(SX_0))Y_1 - (k_{CSA}Y_0 + s)(SX_1)$
2 nd Moments of dormant chains	$\frac{\mathrm{d}(SX_2)}{\mathrm{d}t} = (k_{CSA0}S_0 + k_{CSA}(SX_0))Y_2 - (k_{CSA}Y_0 + s)(SX_2)$
0 th Moments of all chains	$\frac{\mathrm{d}N_0}{\mathrm{d}t} = k_iMC - s(Y_0 + X_0 + SX_0)$
1 st Moments of all chains	$\frac{\mathrm{d}N_1}{\mathrm{d}t} = k_iMC + k_pMY_0 - s(Y_1 + X_1 + SX_1)$
2 nd Moments of all chains	$\frac{\mathrm{d}N_2}{\mathrm{d}t} = k_iMC + k_pMY_0 + 2k_pMY_1 - s(Y_2 + X_2 + SX_2)$

Table 3.3 Mole balance for catalyst, CSA, ethylene, and hydrogen in a dynamic CSTR.



Description	Molar equations
Catalyst	$\frac{dC}{dt} = -(k_iM + k_d + s)C + (k_{i\beta} + k_{tH}H_2 + k_{CSA0}S_0)Y_0 + C^{in}$
Chain shuttling agent (CSA)	$\frac{dS_0}{dt} = -(k_{CSA0}Y_0 + s)S_0 + S_0^{in}$
Ethylene	$\frac{dM}{dt} = -(k_iC + k_pY_0 + s)M + M^{in}$
Hydrogen	$\frac{dH_2}{dt} = -(k_{tH}Y_0 + s)H_2 + H_2^{in}$
Deactivated site	$\frac{dC_d}{dt} = k_d(Y_0 + C) - sC_d$
Ethylene consumption	$\frac{dM_p}{dt} = (k_iC + k_pY_0)M - sM_p$

By solving this set of equations, information about the average chain length can be obtained from Table3.4.

Table 3.4 Average chain lengths

Description	Number average chain length, r_n	Weight average chain length, r_w	Polydispersity, PDI
Overall polymer	$\frac{N_1}{N_0}$	$\frac{N_2}{N_1}$	$\frac{r_w}{r_n}$

Under factory or laboratory conditions, product yield, weight-average molecular weight (Mw) and polydispersity index (PDI) are all measurable. These data can serve as inputs for subsequent PINNs models.

3.3 Inverse PINNs with polymerization model

We propose a deep learning framework based on physics-informed neural networks that incorporates information from the ethylene chain shuttling polymerization mechanism model, which has been developed using the method of moments. After implementing the ODE system and collecting data measurements, we construct our neural network model. The networks input is time t, and the output is a vector of state variables $\hat{\mathbf{x}}(t;\boldsymbol{\theta}) = (\hat{x}_1(t;\boldsymbol{\theta}), \hat{x}_2(t;\boldsymbol{\theta}), \dots, \hat{x}_S(t;\boldsymbol{\theta}))$ which acts as a proxy to the ODE solution.

In the process of training neural networks, it is essential to impose constraints that correspond to the system of ODEs and the previously generated observations. This constraint enforcement is accomplished through the formulation of a loss function, which is responsible for quantifying the disparity between the neural networks' output and the intended behavior： according to the data at the time $t_1, t_2, \dots, t_{N^{data}}$, and ODEs at time points $\tau_1, \tau_2, \dots, \tau_{N^{ode}}$. The selection of time points for ODEs can be chosen at random or uniformly spaced. We establish the aggregate loss as a function of θ and p:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{p}) = \mathcal{L}^{data}(\boldsymbol{\theta}) + \mathcal{L}^{ode}(\boldsymbol{\theta},\boldsymbol{p}) + \mathcal{L}^{e}(\boldsymbol{\theta})\,. \tag{12}$$

For M sets of observations y, $\mathcal{L}^{data}$ is defined:

$$\mathcal{L}^{data}(\boldsymbol{\theta}) = \sum_{m=1}^{M} w_m^{data}\mathcal{L}_m^{data} \tag{13}$$

$$= \sum_{m=1}^{M} w_m^{data}\left[\frac{1}{N^{data}}\sum_{n=1}^{N^{data}}\left(y_m(t_n) - \hat{x}_{s_m}(t_n;\boldsymbol{\theta})\right)^2\right].$$

$\mathcal{L}^{ode}$ is defined for our ODE model:

$$\mathcal{L}^{ode}(\boldsymbol{\theta},\boldsymbol{p}) = \sum_{s=1}^{S} w_s^{ode}\mathcal{L}_s^{ode} \tag{14}$$

$$= \sum_{s=1}^{S} w_s^{ode}\left[\frac{1}{N^{ode}}\sum_{n=1}^{N^{ode}}\left(\frac{d\hat{x}_s}{dt}\,|_{\tau_n} - f_s(\hat{x}_s(\tau_n;\boldsymbol{\theta}),\tau_n;\boldsymbol{p})\right)^2\right].$$

The third term in the loss function is $\mathcal{L}^a$, which serves as an additional informative measure for system identification purposes. For example, here we assume that we have the measurements of all state variables at the initial moment T_0. This one is not necessary, you can choose according to the actual situation.

$$\mathcal{L}^a(\boldsymbol{\theta}) = \sum_{s=1}^{S} w_s^a\mathcal{L}_s^a \tag{15}$$

$$= \sum_{s=1}^{S} w_s^a(x_s(T_0) - \hat{x}_s(T_0;\boldsymbol{\theta}))^2.$$

Then, select the appropriate weight w so that the different loss function terms are of the same order of magnitude. Once the loss functions are established, the networks can be trained to infer the parameters of the ODEs by minimizing the loss function using a gradient-based optimization algorithm such as Adam.

$$\boldsymbol{\theta}^*, \boldsymbol{p}^* = arg\min_{\boldsymbol{\theta},\boldsymbol{p}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{p})\,. \tag{16}$$

During trainning we optimize θ and p simultaneously.

4. SIMULATION

In this section, we utilize the constructed CSA model to obtain simulated information about the average chain length. Subsequently, a parameter identifiability analysis is conducted on the model. Next, we use simulated data to train the Inverse PINNs for identifying kinetics parameters.

4.1 Simulating the method of moments model.

In order to enhance the numerical simulation and decrease the computational cost of the model, the following assumptions were incorporated into the model while adhering to the polymerization reaction kinetics:

1) the immediate activation of the catalyst through a co-catalyst, 2) a consistent reaction volume, 3) a constant polymerization temperature, 4) reaction rates unaffected by chain length, 5) the equivalence of the initiation rate constant, k_i, and the propagation rate constant, k_p, and 6) active sites resulting from β-hydride elimination, chain transfer to hydrogen, and chain shuttling to CSA, all exhibiting analogous behavior to those initially generated during the activation phase. Chain transfer occurs through two mechanisms: β-hydride elimination and chain transfer to hydrogen.

In a 6-hour simulation of a CSTR with a 600-second average residence time, we investigated the effects of changing operational conditions. Tables 4.1 and 4.2 specify kinetic constants and initial conditions. After reaching a steady state, the conditions of the CSTR were modified as indicated in Table 4.3.

Table 4.1 Kinetic constants for ethylene chain shuttling polymerization.

Parameter	Values	Units
k_i	9×10^3	mol/(L·s)
k_p	9×10^3	mol/(L·s)
$k_{t\beta}$	4.5	mol/(L·s)
k_{tH}	500	mol/(L·s)
k_d	0.001	mol/(L·s)
k_{CSA}	8.5×10^4	mol/(L·s)
k_{CSA0}	8.5×10^4	mol/(L·s)

Table 4.2 Initial process conditions for ethylene chain shuttling polymerization in a dynamic CSTR.

Reactants	Values	Units
Monomer molar flow rate, M^{in}	$2/600$	mol/(L·s)
Catalyst molar flow rate, C^{in}	$8 \times 10^{-7}/600$	mol/(L·s)
Chain shuttling agent molar flow rate, S^{in}	$8 \times 10^{-4}/600$	mol/(L·s)
Hydrogen molar flow rate, H_2^{in}	$0.005/600$	mol/(L·s)
Average residence time, τ	600	s

Table 4.3 Process condition changes for ethylene chain shuttling polymerization in a dynamic CSTR.

Time, h	Condition before change (mol/(L·s))	Condition after change (mol/(L·s))
1.5	$M^{in} = 2/600$	$M^{in} = 4/600$
3	$S^{in} = 8 \times 10^{-4}/600$	$S^{in} = 1.6 \times 10^{-3}/600$
4.5	$H_2^{in} = 0.005/600$	$H_2^{in} = 0.01/600$

We utilized MATLAB to solve the system of differential equations established using the method of moments in Table 3.2, 3.3, 3.4 and reaction conditions provided in Tables 4.1, 4.2, 4.3 to determine the production, Mw, and Mn of polyethylene chain-shuttling polymerization. These values serve as observational data for the PINNs. Figure 4.1 illustrates the variations of ethylene, H2, CSA, and Mn over time in a CSTR. All changes reflect the expected trends shown in Table 4.3.

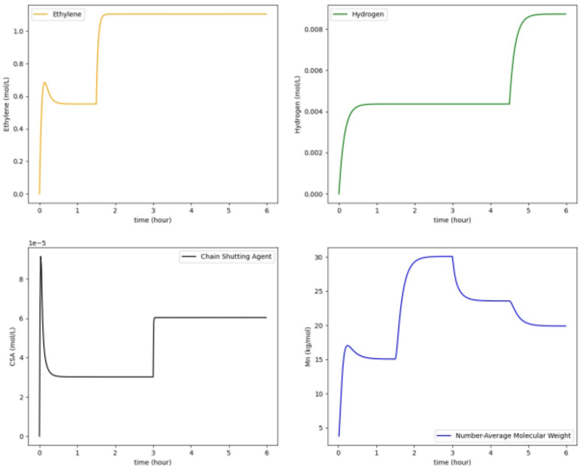


Fig 4.1 Ethylene, H2, CSA, and Mn in a CSTR as a function of time

4.2 Structural identifiability results

In our work, the yield and Mw of polyethylene are considered measurable under factory or laboratory conditions (Hustad, Kuhlman et al. (2008)). In this section, we will investigate whether the kinetic parameters in the polymerization process model established in Section 3.2 can be structurally identifiable from these known inputs (Ovchinnikov, Pogudin et al. 2021).

When knowing Yield, Mw, we can obtain N_1, N_2 through simple calculations. For convenience, we refer to a system as being identifiable when it is structually identifiable. We use the MATLAB toolbox STRIKE-GOLDD to test for parameter structural local identifiability of the model. (Villaverde AF, Barreiro A, et al. 2016) Result shows the ODE system of CSA is structurally identifiable when giving Yield, Mw as observations.

4.3 Use Inverse PINNs for parameter identification

When performing parameter identification, we need to fix the search range of parameters, because we only use the observation of Yield, Mw, based on our structural identifiability analysis. In different problems, the search range of parameters is also different. In this case, the range of parameters is set as $(0.2p^*, 1.5p^*)$ where p^* is the nominal value of that parameter, as illustrated in Table 4.4.

Table 4.4 Parameters for the model

Parameter	Nominal Value	Units	Search range
k_i	1×10^4	mol/(L·s)	$(5\times 10^3, 1.5\times e^4)$
k_p	1×10^4	mol/(L·s)	$(5\times 10^3, 1.5\times e^4)$
$k_{t\beta}$	5	mol/(L·s)	(2.5,7.5)
k_{tH}	500	mol/(L·s)	(250,750)
k_d	0.01	mol/(L·s)	(0.005,0.015)
k_{CSA}	1×10^5	mol/(L·s)	$(5\times 10^4, 1.5\times 10^5)$
k_{CSA0}	1×10^5	mol/(L·s)	$(5\times 10^4, 1.5\times 10^5)$

When applying PINNs, at first step, we define all parameters to be estimated k_i, k_p, $k_{t\beta}$, k_{tH}, k_d, k_{CSA}, k_{CSA0}. To expedite networks training, we can augment the basic FNNs outlined in Section 2.2 with supplementary layers, as delineated below. In situations characterized by an expansive time domain, the variable t exhibits variations spanning multiple orders of magnitude, which can detrimentally impact our neural networks training process. To address this challenge, we employ a linear input scaling layer on t, utilizing the maximum value within the time domain, denoted as T to transform t accordingly: $\tilde{t} = t/T$. As the outputs $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_S$ may have a disparity of magnitudes, we can scale the networks outputs by $\hat{x}_1 = k_1\tilde{x}_1, \hat{x}_2 = k_2\tilde{x}_2, \ldots, \hat{x}_S = k_S\tilde{x}_S$ like in Fig4.2, where $k_1, k_2, \ldots, k_S$ are the magnitudes of the ODE solution $x_1, x_2, \ldots, x_S$ respectively.

The choice of activation function is crucial for the accuracy of PINNs method. In general, it is common practice to choose a higher-order differentiable tanh as the activation function for PINNs because they require computation of higher-order derivatives. However, research(A. Al Safwan, C. Song et al. (2021)) shows that the Swish activation function is superior to tanh in terms of convergence rate for complex problems. The Swish function has a smoother curve shape, which allows for better utilization of gradient information, leading to improved training efficiency and faster convergence of the networks. Furthermore, the Swish function exhibits stronger nonlinear expression capability. In the follow-up experiment, we selected "swish" as the activation function. (激活函数)

During the construction of PINNs , the architecture of neural networks plays a crucial role in determining their performance. Increasing the network's width can enhance its representational capacity. However, excessively large widths may lead to overfitting and inefficient utilization of computational resources. Elevating the depth of neural networks helps in capturing more complex features and representations. However, an excessively deep architecture can increase training complexity and require greater computational resources. When selecting a network structure, it is crucial to strike a balance between expressive power, computational efficiency, and available resources while considering problem complexity and data volume. After many experiments, the neural network has shown more balanced performance when the set size is set to 5*256. (神经网络的结构)

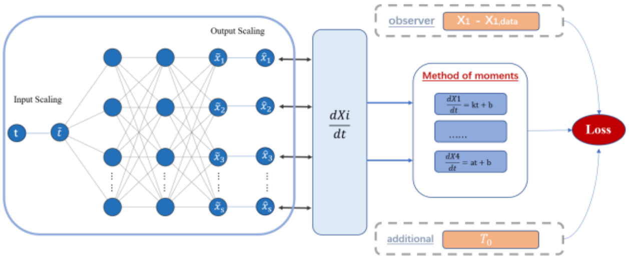


Fig 4.2 Simplified architecture of inverse PINNs for ethylene CSA polymerization in a CSTR model

The algorithm is implemented in Python using the open-source library DeepXDE (Lu, Meng et al. (2021)). The width and depth of the neural networks depend on the size of the system of equations and the complexity of the dynamics.

We randomly sampled 80 points from the simulation results. Additionally, we ensured that the majority of data points were concentrated under steady-state conditions. Moreover, in order to test the robustness of the algorithm, we add random Gaussian noise to the observed data to simulate the measurement error in practice, as shown in the sampled input illustrated in Figure 4.3.

For training purposes, we use the Adam optimizer (Kingma and Ba (2014)) with default hyperparameters and a learning rate set to 10^{-4}. The training process is conducted using the entire dataset as a full batch. Given that the overall loss consists of two supervised losses and one unsupervised loss, we conduct the training using a two-stage strategy as follows:

Step 1. We first train 10,000 epochs by setting all weights to zero except for observation data, to minimize $\mathcal{L}^{data}$ such that the networks can quickly match the observed data points.

Step 2. We further train the networks using all the three losses for 90,000 epochs to minimize total loss function.

Table 4.5 Result of kinetic parameter identification using PINNs

Parameter	Experiment Value	Inferred Value	Error
k_i	9×10^3	8.9×10^3	1.1%
k_p	9×10^3	8.9×10^3	1.1%
$k_{t\beta}$	4.5	4.36	3.1%
k_{tH}	500	518.3	3.7%
k_d	0.001	0.0013	30%
k_{CSA}	8.5×10^4	8.3×10^4	2.4%
k_{CSA0}	8.5×10^5	8.3×10^5	2.4%

After numerous tests, the average running time of PINNs in 100,000 iterations is approximately 3100 seconds. The inferred parameters are provided in Table 4.5. We observe good agreement between the inferred values and the noise-sampled inputs, and Fig4.3 shows the predicted values of Mn and Yield.

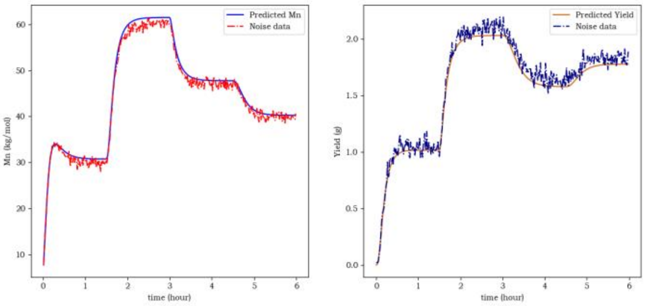


Fig 4.3 PINNs prediction results through noise data

Except for parameter k_d ,which has a slightly larger identification error, the results for the other parameters are accurate, and the prediction curve reflects the changes in polymer properties well.

4.4 Compare PINNs with PSO

PSO is a heuristic optimization algorithm that seeks the best solution by adjusting the position and velocity of particles. In the parameter identification task, PSO can be utilized to search for the optimal parameter combination that minimizes the error between the predicted and observed values.

In this comparison experiment, we used PSO to identify dynamic parameters. Every iteration of PSO needed to call the differential equation model to get the model output, in order to minimize the error between the model output and the observed noise value. Set the dimension of particle swarm to 6, the number of particle population to 500, the maximum number of iterations to 200, and the learning factors to 0.5 and 0.3 respectively. The search ends when the maximum number of iterations is reached or the error is less than 10^{-3}.

When PSO is used to identify all parameters, the error may not converge. After experiments, PSO can only identify three parameters and fix the other parameters to their empirical values. The results are given in Table 4.6.

Table 4.6 Result of kinetic parameter identification using PSO

Parameter	Experiment Value	Inferred Value	Error
k_i	9×10^3	7.8×10^3	13.3%
k_p	9×10^3	7.8×10^3	13.3%
$k_{t\beta}$	4.5	3.22	28.4%
k_{tH}	500	*	*
k_d	0.001	0.0015	50%
k_{CSA}	8.5×10^4	*	*
k_{CSA0}	8.5×10^5	*	*

Table 4.7 Algorithm run time

Methods	Average running time
PINN	3312s
PSO	1106s

The results above indicate that while the PSO algorithm is time-efficient, its recognition success rate and accuracy significantly lag behind those of PINNs, particularly in scenarios with limited observation data and noisy data.

5. CONCLUSIONS

In this paper, we introduce a novel approach for identifying key kinetic parameters in chain shuttling polymerization processes through the integration of the method of moment matching and PINNs. The dynamic variations of the average chain length are characterized using ODEs based on the method of moment matching. The incorporation of PINNs in estimating unknown parameters proves to be advantageous. It efficiently integrates polymerization mechanisms with sparse and limited observational data, thereby reducing the amount of required training data. The comparative analysis between the parameters estimated by PINNs and the ground truth values emphasizes the model's high accuracy and efficiency. Moreover, the paper addresses the challenge of structural identifiability through an identifiability test, demonstrating the adaptability of the proposed method in situations where direct parameter incorporation may not be feasible. This work underscores the potential value of PINNs in chemical process identification, offering a promising avenue for efficient and accurate parameter estimation in complex dynamic systems.

REFERENCES

Bystritskaya, E., et al. (1999). Prediction of the aging of polymer materials. *Chemometrics and intelligent laboratory systems*, 47(2), 175-178.

Chen, Y., et al. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8), 11618-11633.

Cuomo, S., et al. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3), 88.

Daneker, M., Zhang, Z., Karniadakis, G. E., & Lu, L. (2023). Systems biology: Identifiability analysis and parameter identification via systems-biology-informed neural networks. In *Computational Modeling of Signaling Networks* , 87-105, Springer US. New York

Hustad, P. D., et al. (2008). An exploration of the effects of reversibility in chain transfer to metal in olefin polymerization. *Macromolecules*, 41(12), 4081-4089.

Karniadakis, G. E., et al. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422-440.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*,1412.6980.

Lu, L., et al. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM review*, 63(1), 208-228.

Mastan, E. and S. Zhu (2015). Method of moments: A versatile tool for deterministic modeling of polymerization kinetics. *European Polymer Journal*, 68, 139-160.

Ngo, S. I. and Y.-I. Lim (2021). Solution and Parameter Identification of a Fixed-Bed Reactor Model for Catalytic CO2 Methanation Using Physics-Informed Neural Networks. *Catalysts*, 11(11), 1304.

Ovchinnikov, A., et al. (2021). Parameter identifiability and input–output equations. *Applicable Algebra in Engineering, Communication and Computing*: 1-18.

Prata, D. M., et al. (2010). Simultaneous robust data reconciliation and gross error detection through particle swarm optimization for an industrial polypropylene reactor. *Chemical Engineering Science*, 65(17), 4943-4954.

Raissi, M., et al. (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.

Villaverde, A. F., et al. (2016). Structural identifiability of dynamic systems biology models. *PLoS computational biology*, 12(10), e1005153.

Zhang, M., et al. (2010). Modeling of α-olefin copolymerization with chain-shuttling chemistry using dual catalysts in stirred-tank reactors: Molecular weight distributions and copolymer composition. *Industrial & Engineering Chemistry Research*, 49(17), 8135-8146.

Al Safwan, Ali, Chao Song, and U. Bin Waheed. "Is it time to swish? Comparing activation functions in solving the Helmholtz equation using PINNs." 82nd EAGE Annual Conference & Exhibition. Vol. 2021. No. 1. *European Association of Geoscientists & Engineers*, 2021.