Modeling and Control for Renal Anemia Treatment with Erythropoietin Using Physics-Informed Neural Network

by

Zhongyu Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Process Control

Department of Chemical and Materials Engineering
University of Alberta

Abstract

Patients with renal anemia are usually treated with recombinant human erythropoietin (EPO) because of insufficient renal EPO secretion. Clinically, this treatment process is labor intensive. It requires trained personnel to assess monthly Hgb levels, consider intra-patient variability and make adjustments every 2 or 4 weeks based on their experience. The purpose of this paper is to develop decision supporting tools to help medical personnel design optimal treatment plans. The establishment of a good hemoglobin (Hgb) response model is a necessary prerequisite for dose optimization design. First we apply physical-informed neural networks (PINN) to build the Hgb response model under EPO treatment. Neural network training is guided by the physiological model to avoid overfitting problems. During the training process, the parameters of the physiological model can be estimated simultaneously. To handle differential equations with impulse inputs and time delays, we propose approximate analytical expressions for the pharmacokinetic (PK) model and weighted formulations for the pharmacology (PD) model, respectively. The improved PK/PD model was incorporated into PINN for training. Tests on simulated data and clinical data show that the proposed method has better performance than the simple data-driven modeling method and the traditional physiological modeling based on the least squares method. But the original PINN does not allow control input, nor can it receive changing initial states, which indicates the original neural network architecture is not suited for the prediction model. In this context, we apply the new physics-informed neural networks for control (PINNC) which is enhanced with network input interfaces for control actions and initial states. This new structure enables the model predic-

tive control (MPC) design for renal anemia treatment. The generated optimal EPO dosage value can be considered as a decision supporting information.

Acknowledgements

First I would like to express my gratitude to supervisor Dr. Zukui Li. It is impossible for me to finish this work without his guidance. I still remember on Dec 23 2021, Dr. Li had a meeting with me and was willing to give me a chance to join in his team. Thanks to him, I have the opportunity to become a MSc student and start my research under his supervision. He has been providing lots of support for my work including sharing literatures, explaining facts and offering solutions, etc., when I run into a stone wall. His abundant knowledge in optimization is the pillar of my work and also let me learn a lot during MSc study. Even though I make mistakes, he still guides me patiently. There are no words to adequately express my gratitude and appreciation for him.

I would also like to thank all my teachers and staff of University of Alberta. I started my master study in the shadow of covid-19 pandemic. It makes everything rough and difficult. But I can feel everyone of teachers and staff is trying their best to help us, keep us safe and impart knowledge. I owe a great deal of thanks to them for their hard work.

I am grateful to doctoral students in Dr. Li's group as well. They have shared research ideas with me and given me many useful suggestions. I also thank the Natural Sciences and Engineering Research Council of Canada for funding for this research. Last but not least, I must thank my friends and my family. Their help and supports are my motivation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Renal anemia is a kind of disease caused by deficiency of erythropoietin (EPO) secretion by the kidneys due to impaired renal function and some toxic substances in the plasma of uremic patients interfering with the production and metabolism of red blood cells (RBC).[1] The primary function of RBC is to transport oxygen from the lungs to other tissues as well as to transport carbon dioxide back to lungs. Oxygen deficiency will stimulate kidneys to produce endogenous EPO. It is the major regulating agent of erythropoiesis because it can stimulate the proliferation of RBC production. For chronic kidney disease patients, endogenous EPO production is insufficient because of renal inadequacy. The reduced endogenous EPO will cause an insufficient number of RBC and finally cause anemia. Clinical manifestations include palpitations, dizziness, drowsiness, etc.[1] Renal anemia is a common complication of chronic kidney disease and a risk factor for cardiovascular complications. Although hemodialysis can improve patients' clinical symptoms, it cannot cure anemia of patients.

The efficacy of exogenetic EPO in the treatment of renal anemia has been well documented.[2] EPO is a glycoprotein with 165 amino acids. During fetal development, EPO is mainly produced in the liver. After birth, 80% of EPO production occurs in the kidneys. In addition, many organs such as the heart, brain, pancreas and retina also produce and locally release small amounts of EPO. As a cytokine and growth

factor, it can affect many organs. When oxygen delivery to specific cells within the kidney is reduced, secretion of EPO increases while circulating in the plasma and stimulating bone marrow progenitor cells, thereby increasing erythrocyte production.[3] If the increase in erythrocyte numbers relieves the hypoxic signal, EPO expression is downregulated. Despite its clinical effectiveness, there are potential drug-induced risks in patients treated with EPO. Because in practice, clinicians usually adjust the frequency and dose of EPO based on current hemoglobin (Hgb) measurements and previous dosing rules. It requires rich clinical experience for doctors. While a low Hgb level leads to anemia, too high Hgb levels can increase the risk of Hgb variation patterns and even mortality for the patient.[4] Therefore, it is important to develop decision support tools that can help the medical staff determine the appropriate dose and frequency of EPO to maintain the target Hgb level and reduce the cost of treatment.

1.2 Methods Background

Model identification is a traditional concept widely used in science and engineering. The early model establishment is based on the physical principle of process knowledge and domain expertise like Newton's law and Maxwell's equations. The physical principle is represented as mathematical formulations such as algebraic equations, ordinary differential equations and partial differential equations. This physics-driven model is a kind of white box model, through which we can understand every physical quantity of the system. However, many complex systems contain elusive physical laws and they cannot be described by quantitative analysis or even characterized by a suitable choice of variables, which leads to plant-model mismatch. Additional factors including multi-scale modelling, high dimensionality, discreteness, time delay, and uncertainties also cause difficulties and computational bottlenecks for online control.[5] Therefore, because of the plant-model mismatch and low computational efficiency, even when we know the physical models of process systems, it may be hard to apply

the models in online controller design.

With the increase of computing power, the data-driven model becomes popular over the years. Based on learning from training data, appropriate data-driven models like regression, genetic algorithms and artificial neural networks can approximate the hidden relation between inputs and outputs efficiently. In this process, explicit knowledge of the system's physical behavior is disregarded and it is a black-box model lacking interpretability and extrapolation. Besides, the dataset used for the training model is often difficult to collect in engineering. As a result, the trained data-driven model is not accurate and universal for engineering applications.

Recently, the gray box model combining the physics-driven model and data-driven model together attracts researchers' attention.[6] Physical principles like constraints,[7] expert knowledge[8] and model structure[9] are incorporated in data-driven model. This hybrid modeling method improves the interpretability and predictive power of neural networks. Meanwhile, this modified modeling approach shows the potential to be applied in model-based controller design due to its high computational efficiency.

The model-based automatic controller has been adopted in the biomedical field. For example, model predictive controllers have been used successfully in the control of blood glucose in diabetes[10] where patients are simulated by a system of differential equations and the prediction model is described as an autoregressive–moving-average (ARMA) model. The model predictive controller will assess the predicted states and produce a series of optimal control inputs in accordance with control goals over a finite prediction horizon. The first insulin dosage input will be adopted in the system. The preceding processes are then repeated when MPC moves on to the next cycle. This design philosophy can also be applied to renal anemia treatment. But in practice, medical records data contain noise, bias and may be inadequate, which makes data-driven methods like artificial neural networks have underperformed.

Equation 1.1 represents the feedforward function of a basic single hidden layer neural network.[11] W_1, W_2 are weighting matrixes and b_1, b_2 are bias vectors respec-

tively. Function $\sigma(\cdot)$ means activation functions like Sigmoid function and Rectified Linear Unit (ReLU) function.

$$y = \sigma(xW_1 + b_1)W_2 + b_2 \tag{1.1}$$

The network parameter estimation is according to the cost function shown in Equation 1.2

$$\theta^* = \underset{\theta}{argmin}\ J(\theta; Loss^{data}) \tag{1.2}$$

where $\theta = \{W_1, W_2, b_1, b_2\}$ is the set of network parameters, $Loss^{data}$ indicates the residual between training data and prediction data. This optimization is based on gradient descent and backpropagation. The i th iteration of parameter update is described as Equation 1.3

$$\theta^{i+1} = \theta^i - \eta^i \nabla_\theta J(\theta^i; Loss^{data}) \tag{1.3}$$

where η is the learning step size. The parameter estimation results mainly depend on the $Loss^{data}$.

To improve the performance of neural networks, we can take advantage of the underlying known physics information during the training process. Equations 1.4-1.6 show a generic differential equation group

$$N(t, x; u(t, x; \theta)) = 0,\ t \in [0, T], x \in D \tag{1.4}$$

$$I(t_0, x; u(t_0, x; \theta)) = 0,\ x \in D \tag{1.5}$$

$$B(t, x; u(t, x; \theta)) = 0,\ t \in [0, T], x \in \partial D \tag{1.6}$$

where $u(t, x; \theta)$ is the solution that we need, θ is the set of parameters in solution function, t and x are time and position vectors, $N(\cdot)$ implies a general differential operator that may include derivatives, linear and nonlinear terms. $I(\cdot)$ and $B(\cdot)$ indicate the initial and boundary conditions respectively. To calculate the parameters

θ of the solution u, we can consider the residual of this differential equation group as following

$$r_N(\theta) = \int_{[0,T]\times D} (N(t,x;\theta))^2 dt\ dx \tag{1.7}$$

$$r_I(\theta) = \int_D (I(t_0,x;\theta))^2 dx \tag{1.8}$$

$$r_B(\theta) = \int_{[0,T]\times \partial D} (B(t,x;\theta))^2 dt\ dx \tag{1.9}$$

The solution parameters can be obtained based on this optimization problem

$$\theta^* = \underset{\theta}{argmin}\ r_N(\theta) \tag{1.10}$$

$$s.t.\ r_I(\theta) = 0,\ r_B(\theta) = 0 \tag{1.11}$$

The core concept of physics-informed machine learning is to incorporate the residual of physical model $Loss^{model}$ into the cost function 1.12 of neural network

$$\begin{aligned}\theta^* &= \underset{\theta}{argmin}\ J(\theta; Loss^{data},\ Loss^{model}) \\ &= \underset{\theta}{argmin}\ J(\theta; Loss^{data},\ r_N(\theta) + \lambda_1 r_I(\theta) + \lambda_2 r_B(\theta))\end{aligned} \tag{1.12}$$

where λ_1 and λ_2 are weighting parameters. Time derivatives and spatial derivatives in $Loss^{model}$ can be calculated by automatic differentiation. Then, based on the gradient descent algorithm, the network parameters are updated through Equation 1.13 until the end of training.

$$\theta^{i+1} = \theta^i - \eta^i \nabla_\theta J(\theta^i; Loss^{data}, Loss^{model}) \tag{1.13}$$

Figure 1.1 is a simple example[12]. We have a dataset generated by a bifurcation function 1.14.

$$y^2 = x \tag{1.14}$$

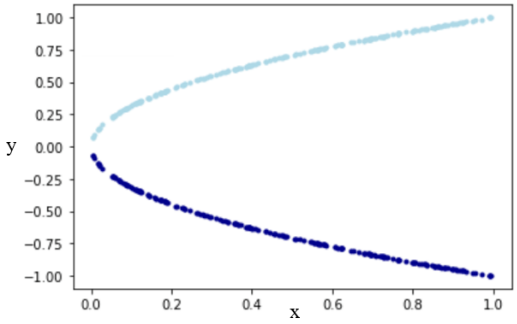
If the training data are axially symmetric like the Figure 1.1a, the pure data-driven neural network with loss function 1.15 will give us an result like the red line in the

Image 1.1b.

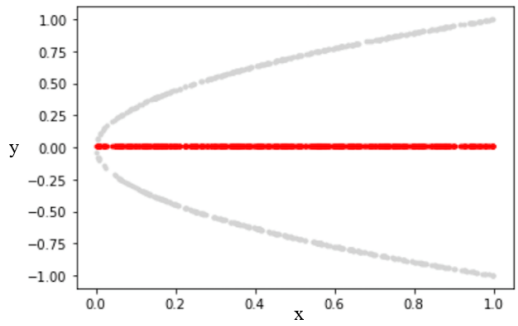
$$Loss^{data} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{net}(x_i))^2 \tag{1.15}$$

With an improved learning setup, we can incorporate the physical model 1.14 into the loss function as Equation 1.16. In this way, we can represent one of the modes of the solution in the plot 1.1c. This simplified example highlights the key capabilities of physics-informed learning approaches. In this thesis, we will embed the established physiological dynamics model with neural networks and implement this combined model in model predictive controller design for renal anemia treatment by EPO.

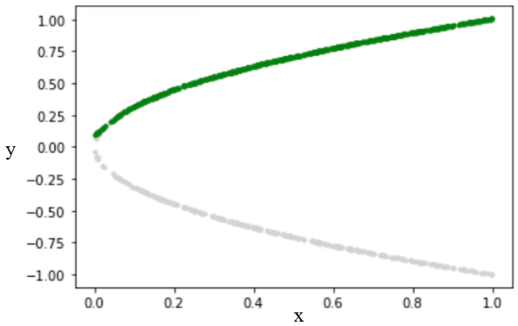
$$Loss^{model} = \frac{1}{n}\sum_{i=1}^{n}(y_{net}(x_i)^2 - x_i)^2 \tag{1.16}$$



(a) Training dataset for pure data-driven neural network



(b) Result of pure data-driven neural network



(c) Result of improved network with physics information

Figure 1.1: Pure data-driven neural network and improved network with physics information

1.3 Thesis Outline and Contributions

The rest of the thesis is organized as follows. Chapter 2 begins with the physiological erythropoiesis modeling used in this work. Then we introduce the basic principle of PINN modeling and propose approximate model equations to the physiological model where impulse input sequence and unknown time delay parameter term exist. After modification of the original physiological model, the PINN method is employed for this problem followed by a test study through a simulation example. Using the proposed approach, we demonstrated the superior performance of the proposed modeling method over pure data-driven autoregressive with extra input (ARX) modeling method and traditional physiological modeling with the least squares method through clinical data. Chapter 3 focuses on the PINNC. We introduce the new network architecture and principle. Self-loop predictions based on PINNC for Van der Pol oscillator system and Hgb response are shown. Chapter 4 is mainly about the classical MPC and zone-MPC results on these two dynamic systems. Finally, in chapter 5, we present future work.

The main contributions of the thesis include:

1) Proposed a method to estimate the parameters of the physiological model using PINN technique

2) Developed a PINNC model structure which combines the physiological modeling and neural network modeling technique, which forms a basis for model predictive control.

3) Proposed a model predictive control framework for optimal anemia treatment through EPO dosage optimization, based on the PINNC model.

Chapter 2

System Identification for Hemoglobin Response Models

2.1 Introduction

Model-based automatic controller design methods have become a popular research area in the biomedical field. A good model for the Hgb-EPO relationship is a necessary prerequisite for feedback controller design. Existing methods of erythropoiesis modeling can be divided into two main categories. One is physiologically driven modeling, which usually uses a combination of pharmacokinetic (PK) and pharmacodynamics (PD) models to describe the dynamics of Hgb concentration following the administration of intravenous EPO.[13–15] Traditional methods such as nonlinear least-squares fitting [16] or genetic algorithms[17] can be used for parameter identification. The other is data-driven modeling, which has been widely applied in medicine[18] like blood glucose dynamics prediction[19], automating administration of hypnotic agents[20] and automated detection of schizophrenia[21]. It sets EPO dose data and Hgb measurements as input and output, respectively. Then through system identification like neural network or autoregressive model methods, we can get the erythropoiesis model.[22, 23] These two methods have their advantages and disadvantages. By building a physiologically driven model, we can get the details of the operation status of the system. Besides, if the theoretical basis for modeling is correct, the physiologically driven model can work well. But in practice, conventional

solvers often have difficulties in finding physiological parameters for ill-posed inverse problems.[24] On the other hand, although a data-driven model can approximate complex functions, it is sensitive to data noise and may not perform in prediction.

Given the above problems, this chapter aims to develop a more efficient method to build an erythropoiesis model with clinical data. The proposed method uses physics informed neural networks (PINN) to identify the model parameters. This kind of neural networks has been applied in many fields like hydromechanics[25] and systems biology[26] and achieved good results. Just like the framework of PINN proposed in paper [27], the front part of the neural network is similar to the ordinary fully connected neural network. After getting the value from the network and calculating the gradient, the physical model equation residuals are incorporated in the loss function to incorporate the physics information. In this way, PINN achieves good estimation and robustness to noise and disturbances.

2.2 Physiological Model

To help physicians make patient-specific decisions on the optimal dosage of EPO treatment, a model that describes the Hgb response to the EPO dosing is necessary.

Regarding the Hgb response to EPO dosage, an example of a clinical data record is shown in Figure 2.1. Hgb level is recorded around every 2 weeks, patients with late-stage renal disease receive EPO treatment 1 to 3 times per week.[28]

Paper [29] has proposed a physiological erythropoiesis model to describe the Hgb-EPO relationship. The model consists of PK and PD model equations. The PK model describes how the body affects specific exogenous chemical substances through absorption and distribution mechanisms after drug administration, as well as changes in the metabolism of substances in the body, and the effects and excretion pathways of drug metabolites. The PD model explains pharmacological effects on living systems, including reactions and binding to cellular components, and the biochemical and physiological consequences of these effects.[30]

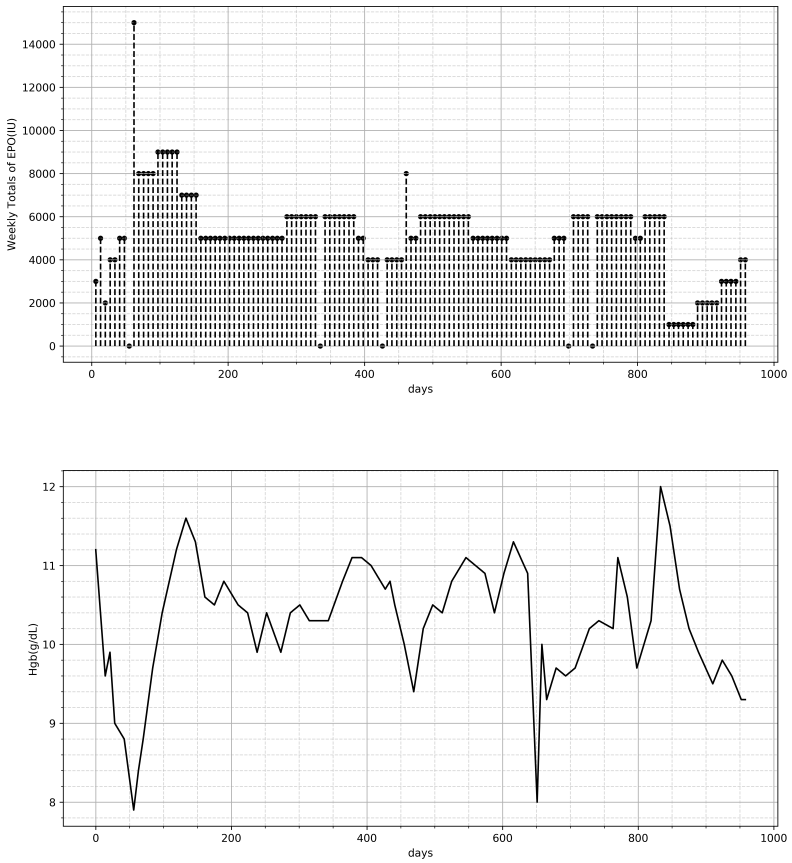


Figure 2.1: A clinical example data of EPO dosages and Hgb responses.

The PK model can be described as:

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\frac{V \cdot E(t)}{K_m + E(t)} - \alpha \cdot E(t) + dose(t) \tag{2.1}$$

$$E_p(t) = E(t) + E_{en} \tag{2.2}$$

$$k_{in}(t) = \frac{S \cdot E_p(t)}{C + E_p(t)} \tag{2.3}$$

$$E_{en} = \frac{C \cdot H_{en}}{\mu \cdot K_H \cdot S - H_{en}} \tag{2.4}$$

The PD model is defined as follows:

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = k_{in}(t-D) - \frac{4x_1(t)}{\mu^2} \tag{2.5}$$

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = x_2(t) \tag{2.6}$$

$$\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} = k_{in}(t-D) - \frac{4x_1(t)}{\mu^2} - \frac{4x_2(t)}{\mu} \tag{2.7}$$

$$Hgb(t) = K_H \cdot R(t) \tag{2.8}$$

In the PK model equations, $E(t)$ denotes the amount of exogenous recombinant human EPO, E_{en} denotes the endogenous EPO, $E_p(t)$ is the total EPO of the dynamic pool in plasma, $k_{in}(t)$ is the red blood cells (RBC) production rate, and $dose(t)$ is the EPO dosing in international unit (IU) which is modeled as a train of impulses.[31] Additionally, the model contains some parameters: H_{en} is the Hgb level due to endogenous EPO, μ represents the mean RBC life span, V is the maximum exogenous EPO clearance rate, K_m stands for the exogenous EPO level that produces half-maximum clearance rate, α is the linear clearance constant, S represents the maximal RBC production rate stimulated by EPO, C is the amount of EPO that produces half-maximum RBC production rate.[28]

In the PD model, states $R(t)$ represent the population of red blood cells (RBC), states $x_1(t)$ and $x_2(t)$ are internal states that aid in calculating $R(t)$, $Hgb(t)$ is the hemoglobin level which can be detected clinically, parameters D is the time required for EPO-stimulated RBCs to start forming, K_H is the average amount of Hgb per RBC (mean corpuscular hemoglobin, or MCH, in a complete blood count) which takes the value of $K_H = 29.5pg/cell$.[28]

The initial conditions can be determined from the data as following

$$R_0 = \frac{Hgb_0}{K_H} \tag{2.9}$$

$$x_{10} = \frac{\mu \cdot (H_{en} - \mu \cdot K_H \cdot \dot{R}_0)}{4K_H} \tag{2.10}$$

$$x_{20} = \frac{K_H \cdot R_0 - H_{en} + \mu \cdot K_H \cdot \dot{R}_0}{K_H} \tag{2.11}$$

where Hgb_0 and $\dot{R}_0$ can be estimated from the data. The prior history of the exogenous EPO is set to be zero. In this PK/PD model, there are eight parameters $(\alpha, C, D, H_{en}, K_m, \mu, S, V)$ to be identified.

Based on the above physiological model, the unknown model parameters can be estimated using the clinical data for each patient. In this work, we use the inverse PINN for the parameter estimation and use the classical nonlinear least squares method with a traditional numerical solver as a contrast.

2.3 Physics-Informed Machine Learning

Paper [27] proposed the PINN which is a type of neural network trained to solve supervised learning tasks while following given physical law described by partial differential equations. It is shown by [32] that the method performs well to identify the unknown model parameters.

The used PINN structure [26] is shown in Figure 2.2. Time is the input. States of the physical system are output. The hidden layers perform nonlinear transformations on the data.[33] It is similar to a fully connected neural network but adds three extra layers to accelerate convergence. Input-scaling layer is designed to shrink the input time domain through a linear scaling function. When differential equations solution has a certain pattern, for example, the solution follows periodicity or attenuation, the feature layer can be set as $sin(kt)$ or e^{-kt} respectively.[32] If states have different magnitudes, the output-scaling layer can be used to scale them.

The main idea of PINN is to incorporate the physical model equation residual (error) into the loss function of the neural network training. During the training process, the network model parameters and the physical model parameters can be estimated simultaneously. Consider a set of ODE equations

$$\frac{\mathrm{d}x_s}{\mathrm{d}t} = f_s(x_s, t; p) \quad s = 1, \cdots, S \tag{2.12}$$

The loss function is composed of three parts as follows.

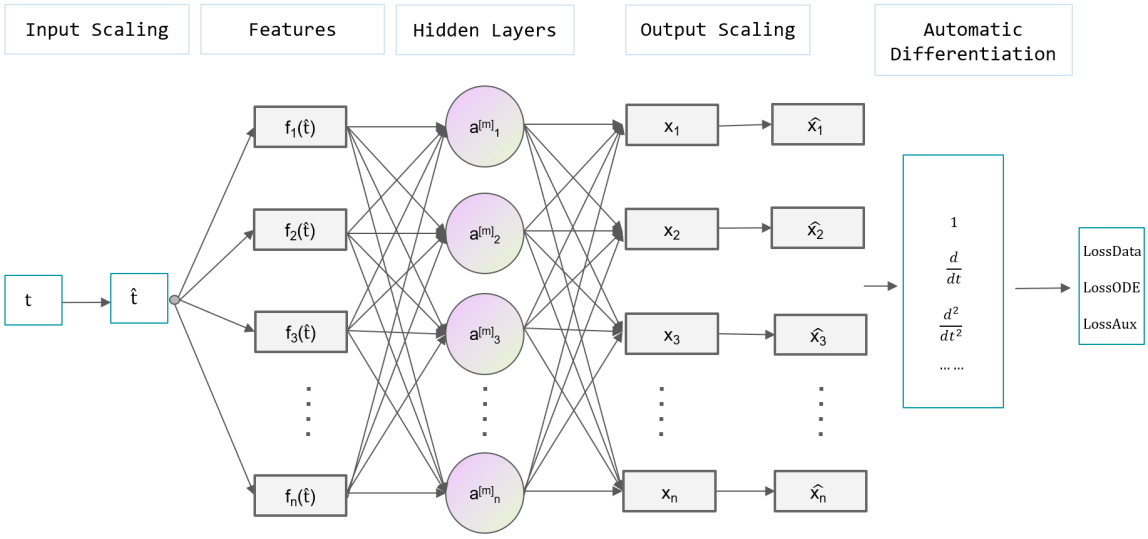


Figure 2.2: Physics-informed neural network architecture

$$Loss(\theta, p) = Loss^{data}(\theta) + Loss^{ode}(\theta, p) + Loss^{aux}(\theta) \tag{2.13}$$

where

$$Loss^{data}(\theta) = \sum_{m=1}^{M} w_m^{data} Loss_m^{data} = \sum_{m=1}^{M} w_m^{data} \frac{1}{N^{data}} \sum_{n=1}^{N^{data}} (y_m(t_n) - \hat{x}_m(t_n;\theta))^2 \tag{2.14}$$

$$Loss^{ode}(\theta, p) = \sum_{s=1}^{S} w_s^{ode} Loss_s^{ode} = \sum_{s=1}^{S} w_s^{ode} \frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left(\frac{\mathrm{d}\hat{x}_s}{\mathrm{dt}}\bigg|_{\tau_n} - f_s(\hat{x}_s(\tau_n;\theta), \tau_n; p) \right)^2 \tag{2.15}$$

$$Loss^{aux}(\theta) = \sum_{s=1}^{S} w_s^{aux} Loss_s^{aux} = \sum_{s=1}^{S} w_s^{aux} (x_s(T_0) - \hat{x}_s(T_0;\theta))^2 \tag{2.16}$$

N^{data} is the number of sample time points $\{t_n\}$ where both the control input and the output response are available. N^{ode} is the number of collocation points $\{\tau_n\}$ used to evaluate the model residual. Note that there is no response data needed for those collocation points. $Loss^{data}$ is based on the errors between measurements of $y_1, y_2, ..., y_M$ and network outputs $\hat{x}_1, \hat{x}_2, ..., \hat{x}_M$ at time $t_1, t_2, ..., t_{N^{data}}$. $Loss^{aux}$ is

similar to $Loss^{data}$, but it specifically considers the start time point T_0 as an additional source. $Loss^{ode}$ is the key point of PINN. By automatic differentiation, the derivative of output states $\hat{x}_1, \hat{x}_2, ..., \hat{x}_S$ concerning input t at the time point $\tau_1, \tau_2, ..., \tau_{N^{ode}}$ can be obtained. Then we can calculate the residual error according to the differential equations and use it as a part of the loss function. In this way, differential equations are integrated into the neural network, which attaches physical constraints to machine learning. The weighting coefficients $(w_1^{data}, w_2^{data}, ..., w_M^{data})$, $(w_1^{ode}, w_2^{ode}, ..., w_S^{ode})$ and $(w_1^{aux}, w_2^{aux}, ..., w_S^{aux})$ are used to balance the loss terms. Finally, by minimizing the loss function, the parameters θ of the neural network and unidentified parameters p of differential equations are optimized together.

2.4 Modified PK/PD Model for PINN

To incorporate the physiological model into the PINN framework, we face two challenges from the original PK/PD model, which are explained below.

2.4.1 Impulse Input Sequences in PK Equation

Equation 2.1 is a differential equation with an impulse input sequence. This equation describes the decay process of exogenous EPO in the human body. Based on a simulation of this differential equation with parameters V, K_m, α being set as $1660, 76.5, 0.25$, respectively, the black dash-dot curve in Figure 2.3 illustrates the trajectory of EPO in the human body during 10 days after receiving 7000 IU EPO medications on the second day.

In practice, $dose(t)$ is a train of impulses. This causes two issues when PINN is used to incorporate this physical equation. First, the width of the impulse tends to be zero and the derivative $\frac{dE(t)}{dt}$ goes to infinity at the dosing time. It is impossible to directly evaluate the differential equation residual $\frac{\mathrm{d}E(t)}{\mathrm{d}t} - \left[-\frac{V \cdot E(t)}{K_m + E(t)} - \alpha \cdot E(t) + dose(t)\right]$. Secondly, the profile of $E(t)$ is not smooth under an impulse sequence input as shown in the figure. It is not very efficient to approximate this nonsmooth function through

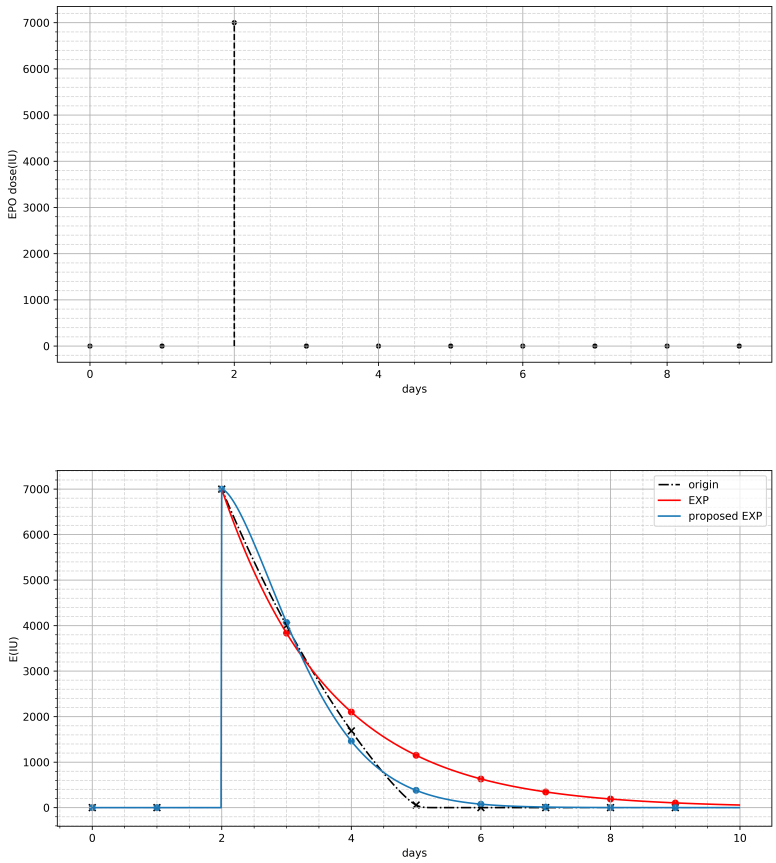


Figure 2.3: Trajectory of $E(t)$ under a single EPO dosage

the neural network. To address this issue, we propose a method to approximate this differential equation based on the following observations.

According to the differential equation 2.1, when $E(t)$ is much bigger than K_m, the equation can be simplified as

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} \approx -V - \alpha E(t) + dose(t) \tag{2.17}$$

For this equation, the solution trajectory of $E(t)$ is an exponential function as shown by the red line in Figure 2.3 with $\alpha = 0.6$. Compared with the exponential function, the curve of differential equation solution $E(t)$ decreases more quickly.

To improve the approximation accuracy, we propose the following exponential func-

tion Equation 2.18 to approximate $E(t)$,

$$E(t) = \sum_{j=1}^{N(t)} dose_j \cdot exp\left[-(a_0 \cdot e^{\frac{-dose_j}{a_1}} + a_2) \cdot (t - t_j)^{a_3}\right] \tag{2.18}$$

where a_0, a_1, a_2, a_3 are four undetermined parameters, t_j and $dose_j$ correspond to the j-th EPO administration time and dosage value, respectively. $N(t)$ is the total number of dosing times up to time tt. For example, if the patient receives 5000 IU EPO treatment and 10000 IU EPO treatment on the 20th day and 60th day, the corresponding (t_j, $dose_j$) are (20, 5000), (30, 10000). Using exponentiation of time difference, this proposed exponential function can match the differential equation solution better at the later stage.

To demonstrate the performance of the proposed model equation, we simulate the original equation 2.1 to get the profile of $E(t)$ under the EPO dose sequence as shown in the top part of Figure 2.3. Afterward, we sample data from the true solution (as shown by the black dash-dot curve in Figure 2.3) and then use the least squares method to estimate the parameters in the proposed model equation 2.18. The estimated parameter values are $a_0 = 1.87, a_1 = 3640, a_2 = 0.269, a_3 = 1.53$. The $E(t)$ trajectory calculated by the proposed exponential function Equation 2.18 is drawn in Figure 2.3 by the blue line, which approximates the true response curve (black dash-dot curve) very well.

In addition to the single impulse input study, the accuracy of the proposed model equation is also tested over a sequence of EPO dosages which was obtained from clinical data. The top part of Figure 2.4 is the EPO treatment record. The solution of the differential equation and the approximate exponential function are shown in the middle part of Figure 2.4, respectively. Notice that the bottom one is the zoomed version of the red box in the middle figure to show more details. R^2 of 958 days $E(t)$ value is equal to 99.76%. This result verifies that the proposed model equation approximates the original exponential differential equation very well.

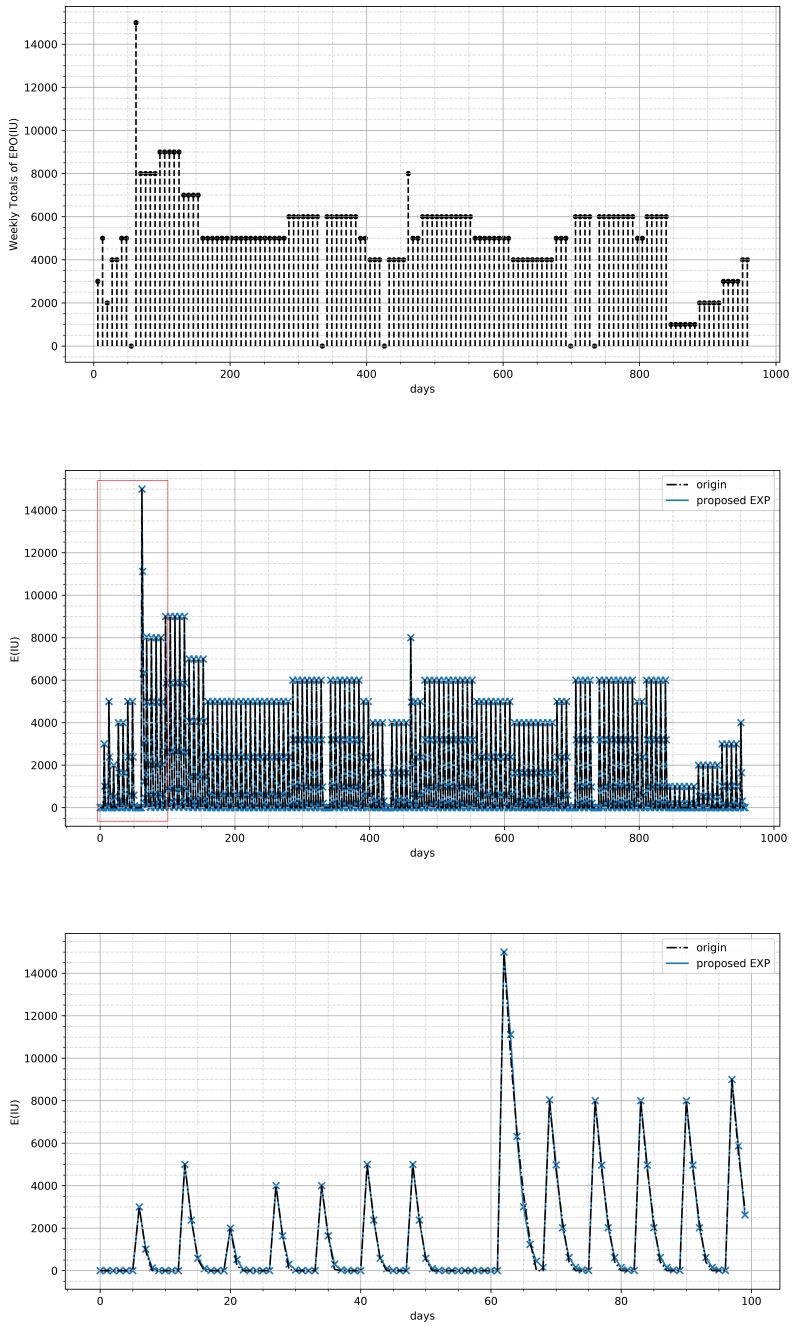


Figure 2.4: Trajectories of EPO in the body based on clinical data

2.4.2 Time Delay in PD Equations

The other issue comes from Equation 2.5 and Equation 2.7. These two equations are delay differential equations. The delay item D is the parameter to be estimated.

Neural networks can predict system states at time t. However, when we calculate the residual of model equations, state prediction at any time is required because parameter D keeps changing with each iteration and it can be any value. But the existing state prediction is limited to the existing input time. So it is not efficient to estimate the delay parameter.

Paper [29] has studied the low-pass filter nature of the RBC pool. As shown in Figure 2.5, a twice-weekly dosing sequence is simulated and it generates pulsatile and periodic EPO levels E_P and corresponding production rate k_{in}; but the periodic dynamics are largely smoothed out by the low-pass nature of the RBC pool filter.[29] Therefore, during the therapy, PK and cell production PD is relevant to the mean value of the production rate, which is denoted as $\bar{k}_{\text{in}}$ in Equation 2.19 where $[iT, (i+1)T]$ is a single dose period. The Equation 2.3 can be reconsidered as a memoryless nonlinear relationship between EPO doses and mean production rate $\bar{k}_{\text{in}}$,[29] which means a similar mean production rate profile $\bar{k}_{\text{in}}$ will lead to similar Hgb trajectory.

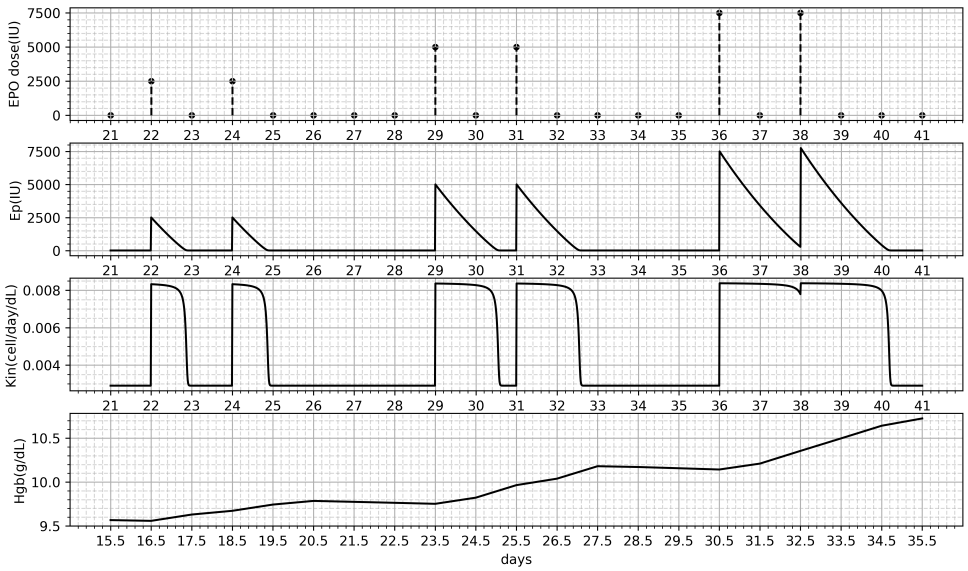


Figure 2.5: Low-pass nature of the RBC pool filter

$$\bar{k}_{\text{in}}(dose_i, T) = \frac{1}{T}\int_{iT}^{(i+1)T} k_{\text{in}}(t)dt \tag{2.19}$$

Based on the above analysis, we propose to convert the delay differential equations into a different form which makes the estimation easier. The method is based on a weighting function and the new equations are defined as follows:

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \lambda_1 k_{in}(t - D_1) + \lambda_2 k_{in}(t - D_2) - \frac{4x_1(t)}{\mu^2} \tag{2.20}$$

$$\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} = \lambda_1 k_{in}(t - D_1) + \lambda_2 k_{in}(t - D_2) - \frac{4x_1(t)}{\mu^2} - \frac{4x_2(t)}{\mu} \tag{2.21}$$

$$\lambda_1 + \lambda_2 = 1 \tag{2.22}$$

The term $k_{in}(t-D)$ is replaced by the weighting function $\lambda_1 k_{in}(t-D_1)+\lambda_2 k_{in}(t-D_2)$, where λ_1 and λ_2 are parameters to be determined, and D_1 and D_2 are fixed as 4 and 7, respectively. This is based on the fact that the time required for progenitor cells to be stimulated by EPO and finally become reticulocytes ready to mature into RBCs is 4 - 7days.[3] The original delay parameter D can be estimated as

$$D = \lambda_1 D_1 + \lambda_2 D_2 \tag{2.23}$$

Figure 2.6 shows the RBC production rate k_{in} and the average weekly production rate $\bar{k}_{\text{in}}(T = 7)$ of the original form and the proposed weighting function respectively. Here, λ_1 and λ_2 are both set as 0.5. It illustrates that during every dose period, there is some difference between the original form $k_{in}(t - D)$ and the proposed weighting function $\lambda_1 k_{in}(t - D_1) + \lambda_2 k_{in}(t - D_2)$. For example, in the second dose period (day 22 to 29), the original model solution includes two pulses whereas the proposed weighting function produces three pulses with a smaller magnitude. However, the average weekly production rate $\bar{k}_{\text{in}}$ of the original form gets superimposed by the $\bar{k}_{\text{in}}$ of the proposed weighting function. The trajectories of Hgb level obtained from these two methods in this short term are similar, as shown in Figure 2.7.

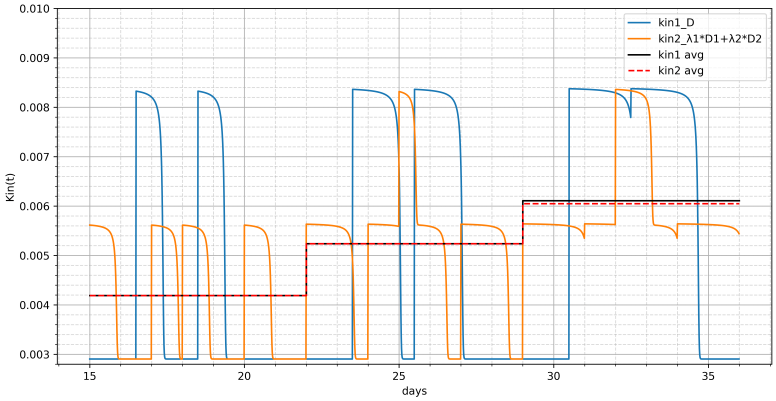


Figure 2.6: Comparison between the average weekly production rate $\bar{k}_{\text{in}}$ and k_{in}

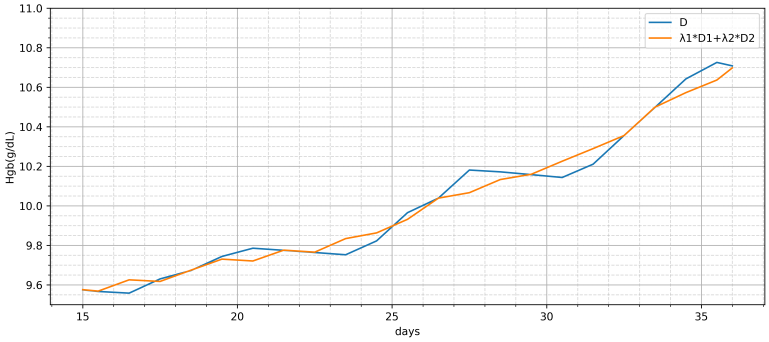


Figure 2.7: Hgb value of two methods in short term

Table 2.1: Parameters for PK/PD models

α	K_m	V	C	D	H_{en}	μ	S
0.25	46.5	2800	22.45	5.5	7.9	92.2	0.0084

Finally, we check the approximation performance over a long horizon. With parameters set as Table 2.1, the original model and the approximated model are both simulated. Figure 2.8 shows these two Hgb trajectories of the original form and delay differential equations with a weighting function. Root mean square error (RMSE) between two curves equals 0.0712.

Above results show that the proposed PK/PD model modification approximate the original model very well. They provide a basis for the PINN modeling as described

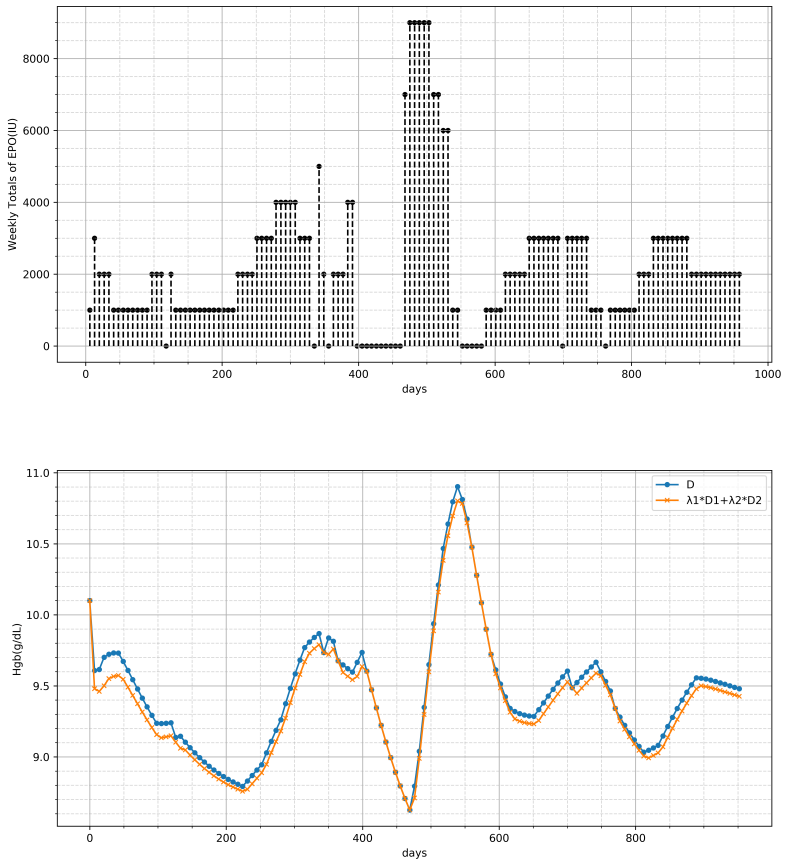


Figure 2.8: Long term Hgb responses: original model and approximation model

in the next section.

2.5 PINN Using the Modified PK/PD Model

According to the proposed approximation model explained in Section 2.4.1- 2.4.2, the overall physiological model used in the PINN is based on equations 2.24-2.31.

$$E(t) = \sum_{j=1}^{N(t)} dose_j \cdot exp\left[-(a_0 e^{\frac{-dose_j}{a_1}} + a_2)(t - t_j)^{a_3}\right] \tag{2.24}$$

$$E_p(t) = E(t) + E_{en} \tag{2.25}$$

$$k_{in}(t) = \frac{SE_p(t)}{C + E_p(t)} \tag{2.26}$$

$$E_{en} = \frac{CH_{en}}{\mu K_H S - H_{en}} \tag{2.27}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \lambda_1 k_{in}(t - D_1) + \lambda_2 k_{in}(t - D_2) - \frac{4x_1(t)}{\mu^2} \tag{2.28}$$

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = x_2(t) \tag{2.29}$$

$$\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} = \lambda_1 k_{in}(t - D_1) + \lambda_2 k_{in}(t - D_2) - \frac{4x_1(t)}{\mu^2} - \frac{4x_2(t)}{\mu} \tag{2.30}$$

$$Hgb(t) = K_H R(t) \tag{2.31}$$

With the above model, the loss function corresponding to the model residual used in the PINN is defined based on equation 2.32.

$$Loss^{ode} = w_1^{ode} Loss_1^{ode} + w_2^{ode} Loss_2^{ode} + w_3^{ode} Loss_3^{ode} \tag{2.32}$$

$$Loss_1^{ode} = \frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left[\frac{\mathrm{d}\hat{R}}{\mathrm{d}t}|_{\tau_n} - (\lambda_1 \hat{k}_{in}(\tau_n - D_1) + \lambda_2 \hat{k}_{in}(\tau_n - D_2) - \frac{4\hat{x_1}(\tau_n)}{\mu^2})\right]^2 \tag{2.33}$$

$$Loss_2^{ode} = \frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left[\frac{\mathrm{d}\hat{x_1}}{\mathrm{d}t}|_{\tau_n} - \hat{x_2}(\tau_n)\right]^2 \tag{2.34}$$

$$Loss_3^{ode} = \frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left[\frac{\mathrm{d}\hat{x_2}}{\mathrm{d}t}|_{\tau_n} - (\lambda_1 \hat{k}_{in}(\tau_n - D_1) + \lambda_2 \hat{k}_{in}(\tau_n - D_2) - \frac{4\hat{x_1}(\tau_n)}{\mu^2} - \frac{4\hat{x_2}(\tau_n)}{\mu})\right]^2 \tag{2.35}$$

During the PINN training process, the neural network parameters and the parameters in the physiological model are simultaneously estimated. Based on the parameters a_0, a_1, a_2, a_3, original parameters V, K_m, α can be further estimated through least squares method. Besides, the delay parameter D can be evaluated using equation 2.23.

Table 2.2: Results: PINN1 (from noise-free data), PINN2 (from noisy data

	α	K_m	V	C	D	H_{en}	μ	S
True	0.1	120	600	30	5.2	9	90	0.004
PINN1	0.0959	127	607	27.6	5.36	8.89	95.0	0.00378
PINN2	0.0966	127	616	30.2	5.38	9.07	84.6	0.00422

2.5.1 Test on Simulated Data

To demonstrate the proposed erythropoiesis modeling based on PINN, true parameter values as listed in Table 2.2 and a set of EPO input sequences as shown in Figure 2.9 are chosen to generate a series of Hgb data by solving this PK/PD model. Then Gaussian noise with zero mean and the standard deviation of $\sigma_\epsilon = c\mu$ is added to Hgb data to simulate measurement noise, where μ is the standard deviation of original Hgb data and c is equal to 5%. According to the noise-free Hgb data and the noise-containing Hgb data, we use PINN to identify these parameters in the differential equations separately and compare the results. The algorithm is implemented in Python with the open-source library DeepXDE.[34] The neural network is formed from 5 fully connected layers and each one has 64, 128, 256, 128, 64 neurons. The feature layer adopts $t, sin(t), sin(2t), sin(3t), sin(4t), sin(5t)$. The tanh function is set as the activation function. In addition, we use the Adam optimizer[35] and 500000 iterations with a learning rate equal to 10^{-4}.

Based on noise-free data and noise-containing data, we can estimate the parameters for erythropoiesis modeling in Table 2.2. The fitting results based on the two cases are shown in Figure 2.9. Corresponding RMSE are 0.0160 (no noise) and 0.0755 (with noise), respectively. The result shows these inferred parameters have a higher degree of accuracy. The agreement between the Hgb solution based on the estimated parameters and exact dynamics is good considering the noise in the training data.

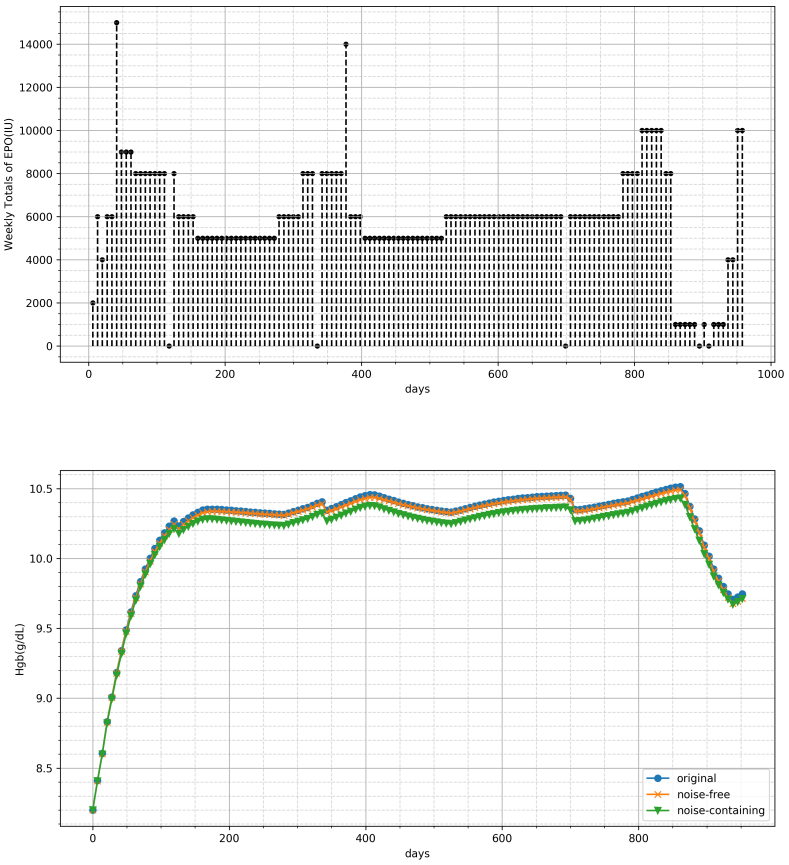


Figure 2.9: Simulated true Hgb responses and PINN model predictions

2.5.2 Application to Clinical Data

Clinical data of 291 patients are used to test the accuracy of the proposed method. General statistics of the clinical data are shown in Table 2.3. After data preprocessing including missing data interpolation, daily Hgb values of patients can be obtained. The first half part of data is used to train and the second half is applied to test the model. The result is compared with ARX modeling method and physiological modeling with the least squares method.

For the physiological modeling with the least squares method, the delay differential equations in the physiological model are solved by dde23 function in Matlab and parameters are optimized using the *lsqnonlin* function in Matlab.[31] The first half part of the data is used to train parameters, and the second half of the data is

Table 2.3: Clinical data statistics

Number of patients	291
Hgb (g/dL) (Mean ± Std)	10.58 ± 1.11
EPO dose (IU/week) (Mean ± Std)	9244.04 ± 10376.34

Table 2.4: Parameters range

Parameter	Nominal value	Unit	Range
μ	77	day	(34, 120)[13]
H_{en}	6.8	g/dL	(4.1, 9.5)[36, 37]
D	5.5	day	(4, 7)[3]
C	22.4	IU	(6.72, 38.08)
V	1655	IU/day	(496.5, 2813.5)
α	0.25	IU/day	(0.075, 0.425)
K_m	76	IU	(22.8, 129.2)
S	0.01	cell/day/dL	(0.003, 0.017)

tested for model error. This parameter estimation based on the least squares is time-consuming because it involves the solution of the delay differential equations many times.

For the proposed method based on PINN, the network structure and optimization are the same as the example in the test on simulated data. Part of the parameters is limited in the physical significance range as shown in Table 2.4. The range for other parameters which do not have reported values in relative literature is set as $(0.3\bar{p}, 1.7\bar{p})$, where $\bar{p}$ is the nominal parameter value. In this method, the training and test rates are set at 50% separately. The number of collocation points N^{ode} in Equation 2.15 is set as $10N^{data}$.

Figure 2.10 show the Train RMSE distribution and Test RMSE distribution of ARX modeling, least squares modeling and proposed PINN modeling based on all clinical data.

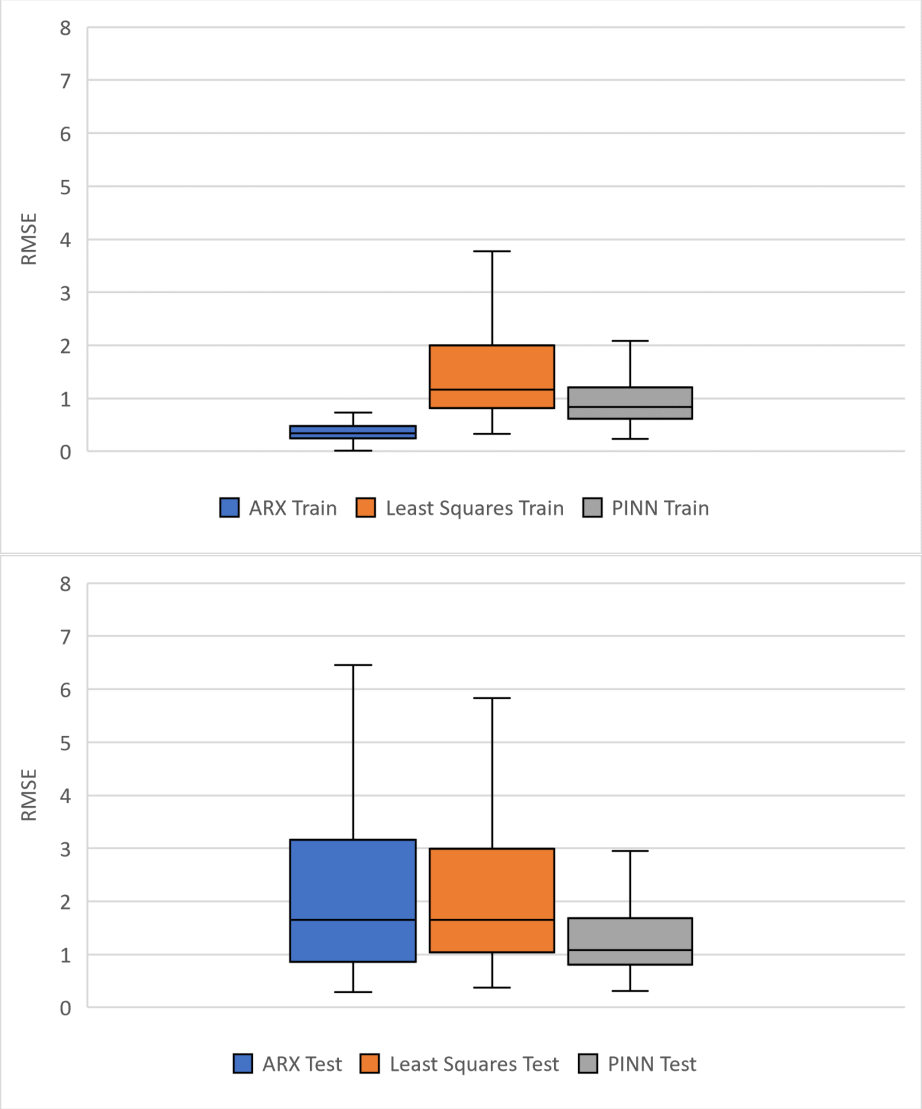


Figure 2.10: Training and test RMSE distribution on clinical data

There are 26 outliers in ARX test RMSE box plot, 17 outliers in the least squares Test RMSE box plot, and 13 outliers in PINN Test RMSE box plot, which is caused by the poor model fit. It can be seen in most cases the proposed PINN modeling method is more accurate and its test error distribution is more concentrated. The proper training error shows it overcomes overfitting and underfitting. This observation can be explained through the following example patients.

For example, the fitting results of three patients are displayed in Figures 2.11, 2.12 and 2.13.

Patient's imitative effect in Figure 2.11 is overall good. The training set is well-fitted and the obtained parameter model of the least squares method and PINN method can predict the test set accurately. The consistency between EPO records and clinical Hgb data shows there is no abnormal situation during the treatment of patient 1, which allows the data-driven ARX model to also perform well

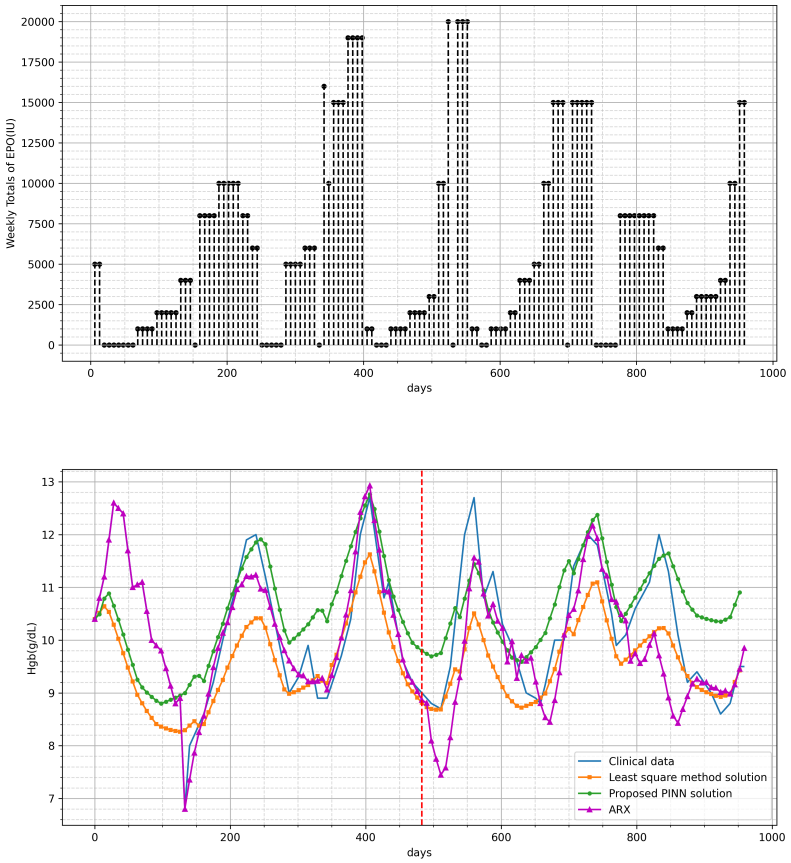


Figure 2.11: Fitting result of patient 1

The record of patient 2 is shown in Figure 2.12, from the 20th day to the 70th day, there is a rapid increase of clinical Hgb data with a series of relatively low-level EPO doses, which is a kind of exceptional situation. Besides, as this patient is treated with very high dosages of EPO from the 200th day to the 240th day, instead, the clinical Hgb record decreases during this period. It can be inferred that there are blood transfusions, internal hemorrhages or similar situations in the therapy of patient 2.

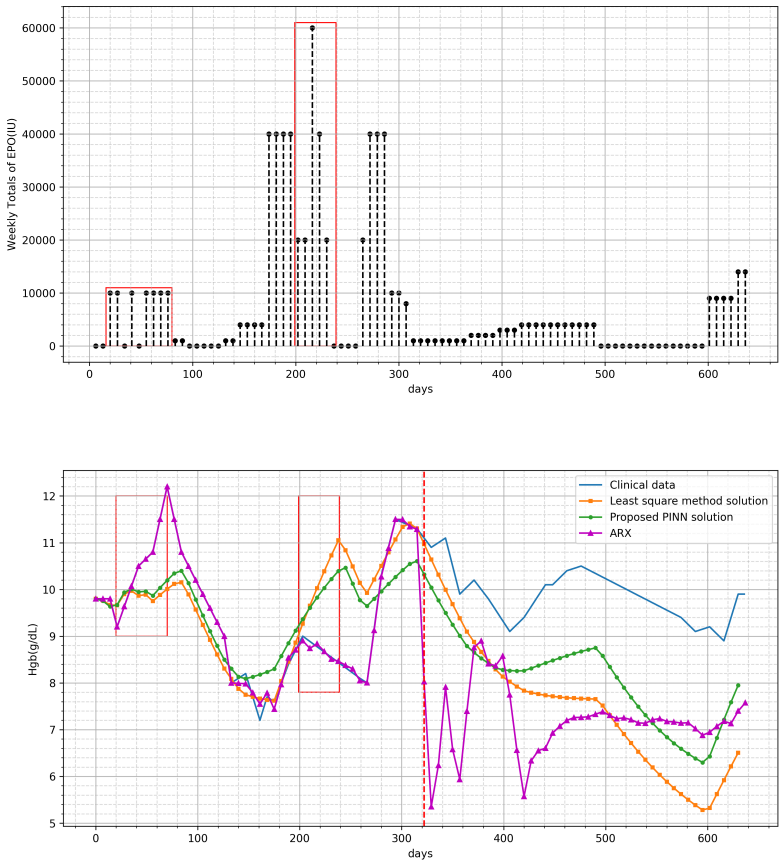


Figure 2.12: Fitting result of patient 2

Figure 2.13 shows that this patient receives similar dosages of EPO medications from the 420th day to the 480th day. The recorded Hgb level keeps increasing in the beginning but drops sharply around day 460, which proves there is a loss of blood. These abnormal conditions are not under consideration in the PK/PD model equations 2.1- 2.8. There is no doubt that if the abnormal data is fitted during training, it will have negative effects on the model parameters and then influence model generalization, which happens in the data-driven ARX methods as well as least squares modeling methods. Our results indicate the physical information allows the neural network to avoid overfitting and have better generalization ability. The noise of measurement and abnormal data caused by intercurrent events are not fitted, which improves the precision of the test set.

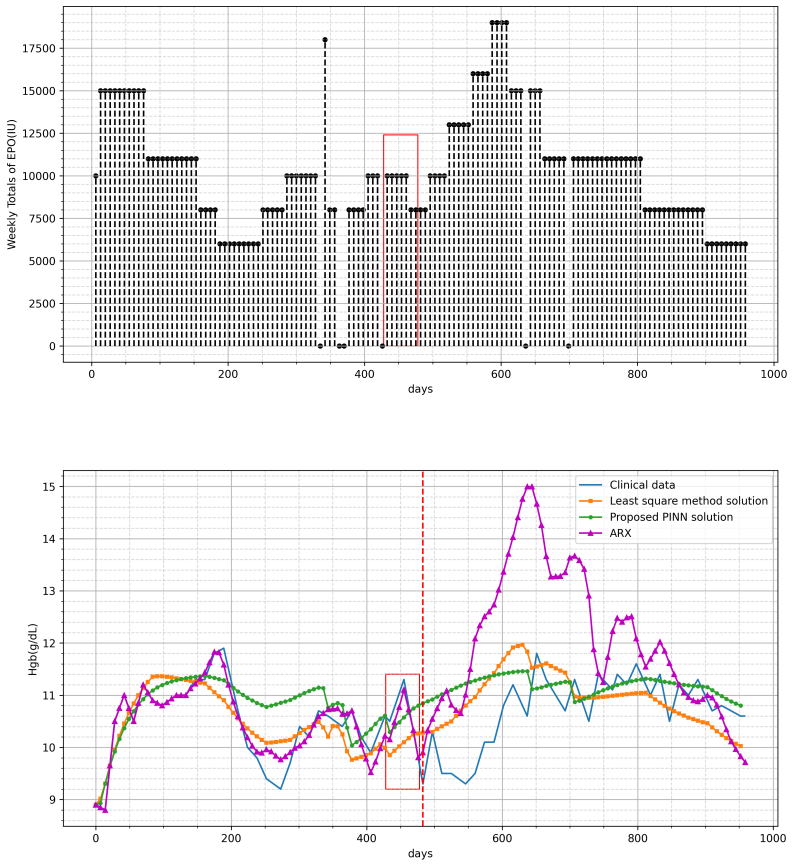


Figure 2.13: Fitting result of patient 3

2.6 Conclusions

In this chapter, we applied PINN technique to model the Hgb response under EPO treatment. This method combined the physiological PK/PD model and neural network learning technology to estimate the parameters of PK/PD model while training the neural network model. During the training of the neural network, physical laws describing the physiological model are enforced by adding the model residuals to the loss function. To address the problem that the PINN cannot easily handle the residual of the differential equation at the time instants with impulse inputs, we proposed an approximate model to replace the PK model equation. In addition, to handle the time delay in the PD differential equation, we proposed a weighting function-based

formulation so that the delay parameter can be estimated by training the PINN. Tests have been made on both simulated data and clinical data. The results show that the proposed method has good accuracy and resistance to overfitting. This is due to the incorporation of the physiological model into the neural networks.

The proposed modeling technique can help build an individualized model for patients with renal disease. Physicians can rely on this modeling technique to develop patient-specific EPO dosing strategies to optimally manage the Hgb level of different patients.

Chapter 3

Physics-Informed Neural Networks for Control

3.1 Introduction

In this chapter, we propose to use a modified PINN in the model predictive controller design for EPO treatment. Model predictive control (MPC) is a feedback control technique that can handle constraints in complex systems and help us to operate the systems optimally. Ordinary differential equations or partial differential equations are the main ideas to simulate dynamic systems. But sometimes the computational cost of solving the differential equations is too high. Even with the improvement in computing power, the efficient real-time model is still a major bottleneck limiting the application of MPC. Currently, there are many data-driven system identification techniques that can replace physical equations. Among these data-driven methods, neural networks are most popular in recent years. Inspired by biological neural systems, it is a powerful function approximator for simulating complex dependencies between inputs and outputs. Especially recurrent neural network (RNN) and its variants in Figure3.1 have good capabilities in handling sequence data.

Due to its novel capability for modeling complex dynamical systems, properly designed artificial neural networks can serve as goal-seeking computational models for solving various optimization problems in many applications,[39] including steel pickling process control[40] and stochastic model predictive control for oil and gas

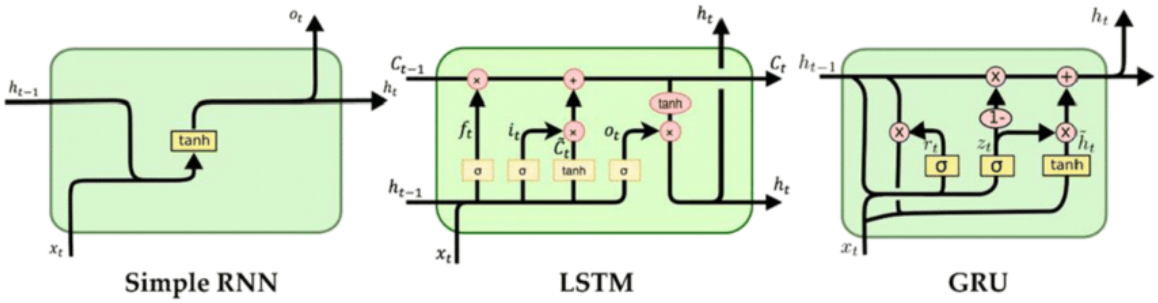


Figure 3.1: RNN and its variants[38]

production[41].

But just like every data-driven method, traditional neural networks do not perform well when facing noisy data and insufficient training datasets. As a result, an inaccurate neural network model will have a destructive impact on model control. For this problem, we propose to use PINN to replace physiological models and traditional neural networks. The detail of PINN has been discussed in chapter 2. The physics information implemented by the loss function can regularize the learning process, which improves the performance of neural networks. But as shown in Figure 2.2, the original PINN does not have an interface for control input. Besides, the initial states are fixed in one value, which does not meet the requirement for rolling optimization in MPC. Considering these limitations, we apply a modified PINN called Physics-Informed Neural Nets for Control (PINNC)[42] in the controller design. The inputs of PINNC are composed of time, initial states and control input. This architecture makes PINNC applicable for MPC implementation. We demonstrate the efficiency of this framework through Van der Pol system and EPO treatment system.

3.2 Physics-Informed Neural Networks for Control Architecture and Prediction

A good prediction model is necessary for controller design. The PINN discussed in chapter 2 is the proper approximation model. The residual of differential equations is added to the loss function to make the network more robust. But the original PINN

can not be applied in MPC. We have to modify its architecture.

3.2.1 Network Structure and Training

A simplified PINN structure for ordinary differential equation is shown in Figure 3.2. The network input is time t, which indicates this neural network represents the function 3.1. The control action and initial conditions are fixed. As a result, the original PINN can infer the unknown parameters in model equations and is suitable for interpolation prediction. But it can not realize extrapolation prediction using extra input. As a consequence, it is not qualified for MPC application. The modified PINNC was proposed in paper [42]. Figure 3.3 explains the network structure. Compared with original MPC, the modified PINNC has two more inputs for control action u and initial conditions. As initial conditions and control actions change with each rolling optimization, the proposed PINNC can generate model predictions based on these inputs. The output of this network is given by Equation 3.2.

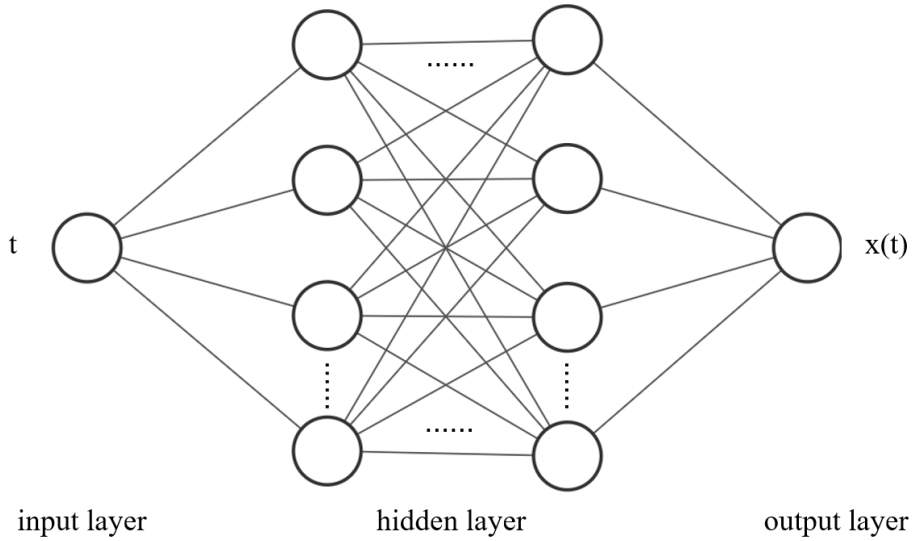


Figure 3.2: Original PINN structure

$$x(t) = f(t) \tag{3.1}$$

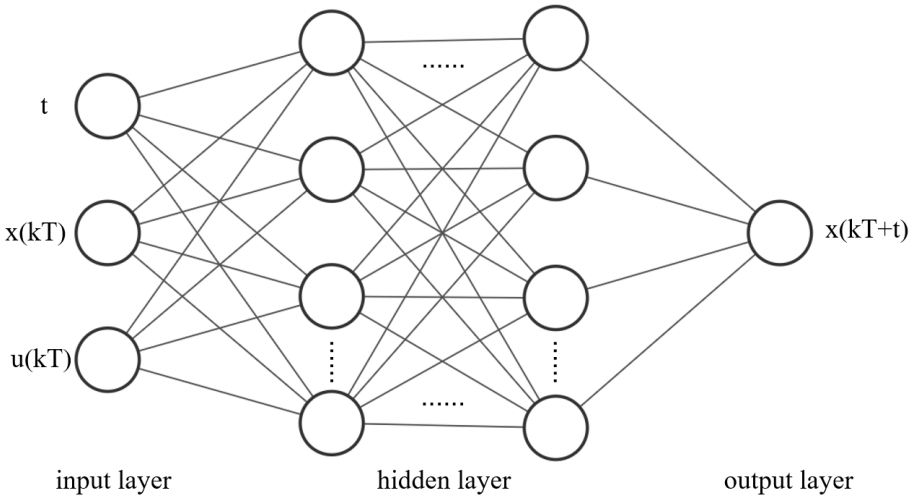


Figure 3.3: PINNC structure

$$x(kT+t) = f(t, x(kT), u(kT)),\ t \in [0, T] \tag{3.2}$$

It's worth noting that in PINNC, we consider multiple equidistant time intervals, each interval is represented by $[kT, (k+1)T]$. Traditional PINN tends to degrade rapidly for long time intervals and can only accept input t in the time duration of the training data.[42] Through the shorter period of T, PINNC solves this degradation problem. Given the initial condition $x(kT)$ and input $u(kT)$, all the states during this time interval can be predicted by trained PINNC. After getting $x((k+1)T)$, we can use the final states as new initial states for the next time interval and repeat this process iteratively.

Furthermore, for the differential equations with input delay defined by Equation 3.3, not only the current input but also these previous control actions are required. In this case, the control input of previous periods is also added to the neural network. The network architecture is given in Figure 3.4.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t, x(t), u(t), ..., u(t-\tau T)) \tag{3.3}$$

The loss function is composed of two parts as follows.

$$Loss(\theta, p) = Loss^{data}(\theta) + Loss^{ode}(\theta) \tag{3.4}$$

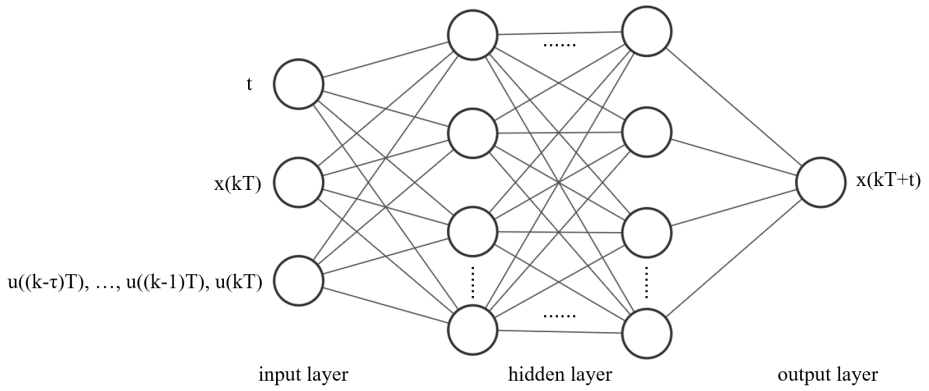


Figure 3.4: PINNC architecture with two inputs

where

$$
\begin{aligned}
Loss^{data}(\theta) &= \sum_{m=1}^{M} w_m^{data} Loss_m^{data} \\
&= \sum_{m=1}^{M} w_m^{data} \frac{1}{N^{data}} \sum_{n=1}^{N^{data}} (y_m(t_n) - \hat{x}_m(t_n, x_n, u_n; \theta))^2, \ t_n \in [0, T] \qquad (3.5) \\
Loss^{ode}(\theta) &= \sum_{s=1}^{S} w_s^{ode} Loss_s^{ode} \\
&= \sum_{s=1}^{S} w_s^{ode} \frac{1}{N^{ode}} \sum_{n=1}^{N^{ode}} \left(\left. \frac{\mathrm{d}\hat{x}_s}{\mathrm{dt}} \right|_{\tau_n} - f_s(\hat{x}_s(\tau_n, x_n, u_n; \theta), \tau_n) \right)^2, \ \tau_n \in [0, T] \\
& \qquad (3.6)
\end{aligned}
$$

In general, the loss function of PINNC is similar to that of PINN. The difference is that in PINNC, the value of time input t is in the time interval, instead of the whole duration of the training dataset. x_n and u_n are corresponding initial states and control input. Besides, the auxiliary loss function is not included because we do not need to enforce one fixed initial condition. The training process is also the same as PINN. A related study has shown that, if the differential equations can represent the processing system accurately, it is sufficient for the training dataset to contain only the initial conditions.[42] But taking full advantage of additional data into loss function can accelerate convergence and improve the accuracy of neural networks. Automatic differentiation is also employed for calculating the residuals of model equations.

3.2.2 PINNC in Self-Loop Prediction

Before applying the PINNC into MPC, we need to ensure the PINNC approximation model is sufficiently accurate. Based on a series of given input, the PINNC can operate as self-loop prediction as Figure 3.5.

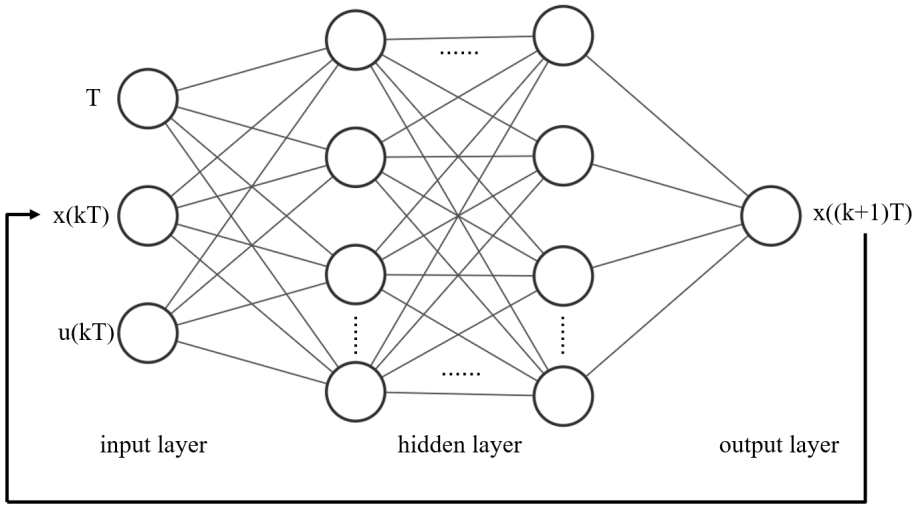


Figure 3.5: Self-loop structure

The initial state for network input can be either the true value measured from the processing system or the model prediction of the previous time interval. In the self-loop prediction, we do not use any true data from the process, except the first initial state, to test the accuracy of PINNC over a prediction period. The self-loop function can be described as follows:

$$\hat{x}((k+1)T) = f(T, \hat{x}(kT), u(kT)),\ k \in N \tag{3.7}$$

The first initial state $x(0)$ is given. Time t in Equation 3.7 is set as the time interval value T, which depends on the dynamics of the process system. One forward propagation is operated every T period. We can see this recurrent approach is different from traditional numerical integration methods that need to calculate integration over the continuous inner interval.[43] Inevitably, recursion calculation will cause error accumulation. But after implementing the PINNC into MPC, we will update the

initial states according to the plant model results. In consequence, the Equation 3.7 can be replaced by the following equation:

$$\hat{x}((k+1)T) = f(T, x^{plant}(kT), u(kT)),\ k \in N \tag{3.8}$$

where the x^{plant} describes the plant model states. The error accumulation is largely eliminated.

3.3 Test of PINNC Modeling Performance

3.3.1 Van der Pol Oscillator

Physicist Balthasar van der Pol proposed the Van der Pol oscillator.[44] At first it was introduced to describe the limit cycle in electrical circuits. Later it was widely used in both the physical and biological sciences. We can use this system as an illustration example. The Van der Pol oscillator equations are defined as follows:

$$\dot{x_1} = x_2 \tag{3.9}$$

$$\dot{x_2} = \mu(1 - x_1^2)x_2 - x_1 + u \tag{3.10}$$

where x_1 and x_2 are two system states, u is referred to the control input, μ represents the damping parameter. Then we solve these ordinary differential equations based on a group of randomly generated input. This dataset can be applied to train the PINNC model. In our training, parameter μ is set as 1, control input belongs to [-1, 1] and generates 50 seconds dataset. Usually, N_{ode} is bigger than N_{data}, which indicates we adopt more collocation points than data points. A sufficient number of points will allow the neural network to perform better. But considering this example is simple, we use $N_{ode} = N_{data} = 100$. Time interval T is equal to 0.5s. For the hyperparameters, the learning rate is 0.001, Adam is used to optimize the loss function, the neural network has 4 hidden layers of 20 neurons each. After 50000 epochs, we get the proper PINNC model. Based on this model, the self-loop prediction on training data can be seen in Figure 3.6. Then we create another control input u series for 25s randomly

and generate a test set by solving the Van der Pol oscillator. The comparison of the test set and PINNC self-loop prediction is shown in Figure 3.7. Although prediction error is accumulated during recursion calculation, the prediction trajectories match the test set very well.

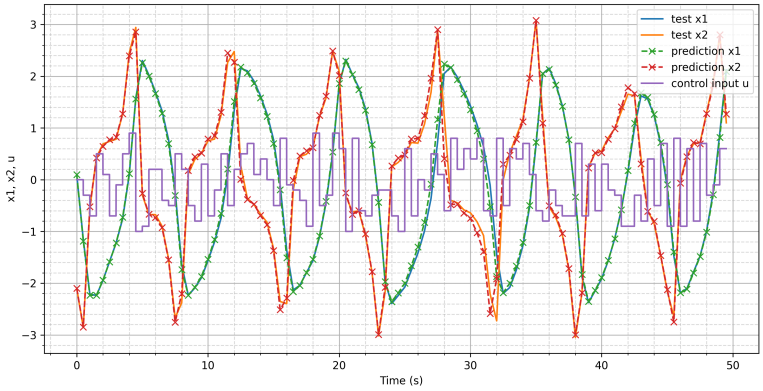


Figure 3.6: PINNC prediction for the Van der Pol oscillator on training set

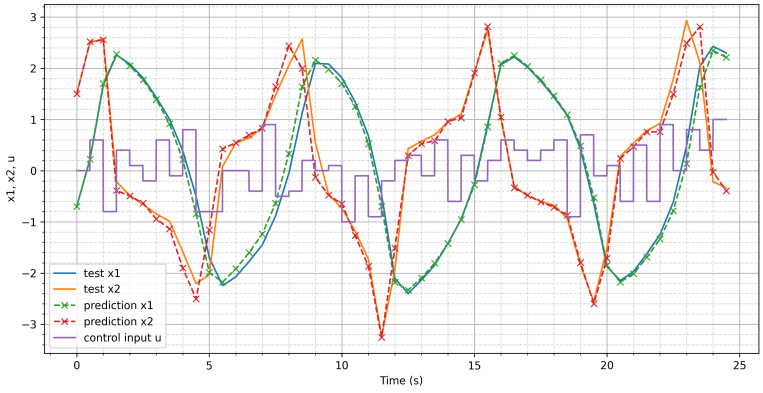


Figure 3.7: PINNC prediction for the Van der Pol oscillator on test set

In addition, we also compare the PINNC model with pure data-driven neural networks. During another training, N_{ode} is set as 0, which implies the residual of physiological model $Loss^{ode}$ is not considered. After the same training, we obtain a pure data-driven neural network. Facing the same test set, the pure data-driven neural network generates prediction as Figure 3.8.

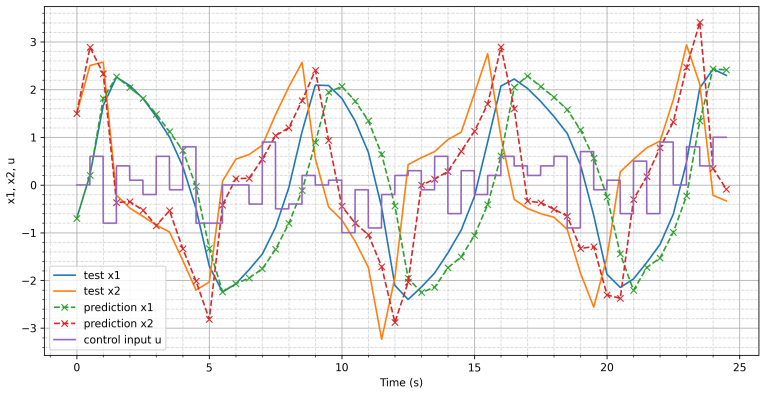


Figure 3.8: PINNC model (with zero weight on model residual) prediction for the Van der Pol oscillator on test set

Table 3.1: Parameters for patients' original model

α	K_m	V	C	D	H_{en}	μ	S
0.384	129	497	38.0	5.61	6.15	90.4	0.00565

In Figure 3.8, the prediction can fit the test set in the beginning. Nevertheless, with error accumulation, the problem that data-driven model precision is lower than PINNC is exposed.

3.3.2 Hgb Response Prediction

Hgb response under EPO treatment is a more complex system. The establishment process of PINNC for this model is similar to that of Van der Pol oscillator system. Through discussion in chapter 2, we apply original PINN to infer the unknown parameters of each patient's physiological model. One patient is picked to illustrate the proposed method. The physiological model and corresponding parameters are defined as Equations 2.1-2.8 and Table 3.1. In this experiment, we solve these equations based on clinical EPO dosages records to generate the training set. Using the trained PINN model to get the training data is another feasible method.

When building the loss function of PINNC based on the original form 2.1-2.8, we will face the same questions mentioned in chapter 2. Impulse input differential

Table 3.2: Parameters for patients' modified model

a_0	a_1	a_2	a_3	C	D	H_{en}	μ	S
2.5	2760	0.0572	1.73	38.0	5.61	6.15	90.4	0.00565

Equation 2.1 makes it impossible to directly calculate the residual and it is not efficient to approximate this nonsmooth function through neural networks. On the other hand, the delay parameter is a fixed value instead of a changing variable. So we do not need to change the delay differential equations into weighting forms again. As a result, the modified physiological equations used for calculating the model loss function of PINNC are Equation 3.11 to Equation 3.18.

$$E(t) = \sum_{j=1}^{N(t)} dose_j \cdot exp\left[-(a_0 e^{\frac{-dose_j}{a_1}} + a_2)(t - t_j)^{a_3}\right] \tag{3.11}$$

$$E_p(t) = E(t) + E_{en} \tag{3.12}$$

$$k_{in}(t) = \frac{S \cdot E_p(t)}{C + E_p(t)} \tag{3.13}$$

$$E_{en} = \frac{C \cdot H_{en}}{\mu \cdot K_H \cdot S - H_{en}} \tag{3.14}$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = k_{in}(t - D) - \frac{4x_1(t)}{\mu^2} \tag{3.15}$$

$$\frac{\mathrm{d}x_1(t)}{\mathrm{d}t} = x_2(t) \tag{3.16}$$

$$\frac{\mathrm{d}x_2(t)}{\mathrm{d}t} = k_{in}(t - D) - \frac{4x_1(t)}{\mu^2} - \frac{4x_2(t)}{\mu} \tag{3.17}$$

$$Hgb(t) = K_H \cdot R(t) \tag{3.18}$$

The required parameters in modified equations are also inferred during chapter 2 as Table 3.2.

In view of the time delay in equations, the EPO input of the last period is added to the neural network. The network architecture in Figure 3.4 is applied. τ is equal to 1. Considering the Hgb treatment process, we set the sampling period T as 7 days. The neural network has 5 hidden layers, and each layer has 128, 256, 512, 256, 128 neurons. Moreover, $N_{data} = 945$. $N_{ode} = 9450$. Adam optimizer is used and the

learning rate equals 0.0001. After 50000 epochs of training, we obtain an appropriate PINNC model for the renal anemia treatment system. Figure 3.9 illustrates the self-loop PINNC prediction on the training set. Besides, multiple test sets are applied to check the performance of this neural network. Firstly, we halve the EPO dosages in the training set and compare the model self-loop prediction with the solution of the original PK/PD model. Figure 3.10 shows the result.

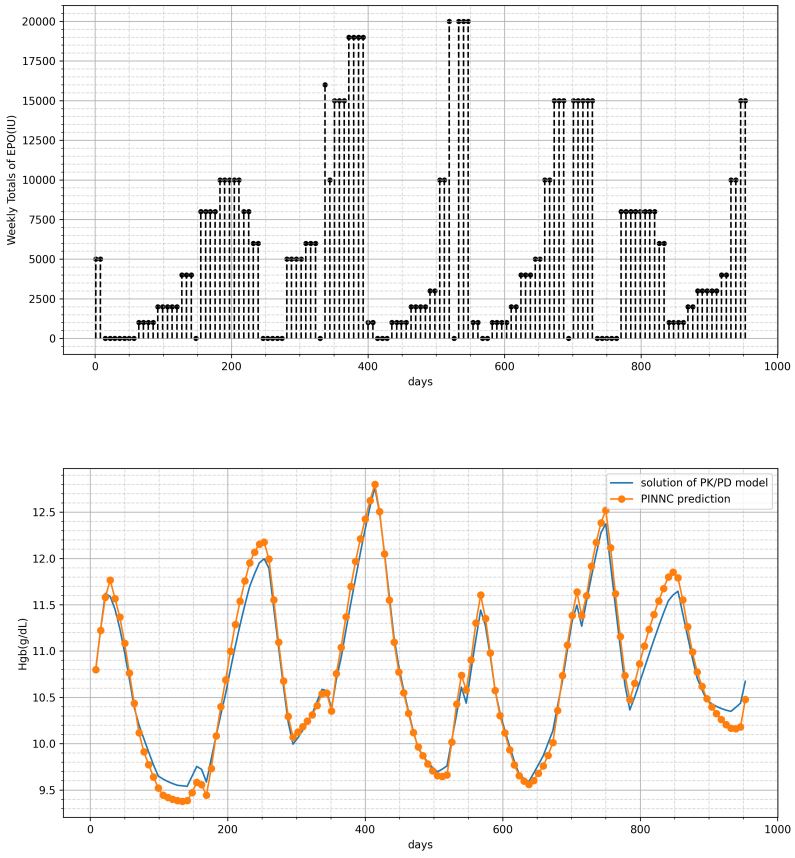


Figure 3.9: PINNC prediction based on training EPO dosages

Then another patient's EPO dosages records are used to test this PINNC. Comparison is explicated in Figure 3.11.

Finally, we calculate the mean value and standard deviation of EPO training data. And in light of the mean value and standard deviation, a group of EPO dosages is generated by Gaussian distribution, which is also put into the neural network as

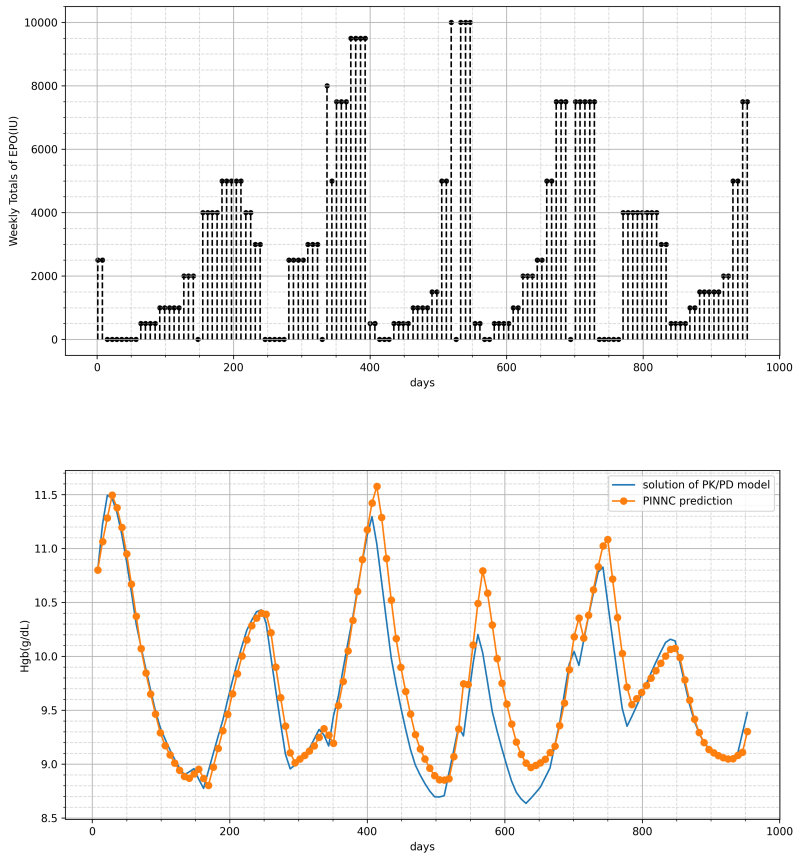


Figure 3.10: PINNC prediction based on half EPO dosages

a control action. In Figure 3.12, these two Hgb trajectories based on this random control input are shown.

In general, the PINNC simulation matches the test set very well. Clearly, there is error accumulation. But in the MPC context, the solution of the optimal control problem is robust regarding this error for larger time horizons, such that this PINNC approximation model is acceptable within the MPC framework.[45]

3.4 Conclusions

In this chapter, we discuss the modified PINN structure for the application in MPC. The original PINN can not meet the requirement to update the initial states and control input. Therefore, we propose to use the PINNC to replace the dynamics predic-

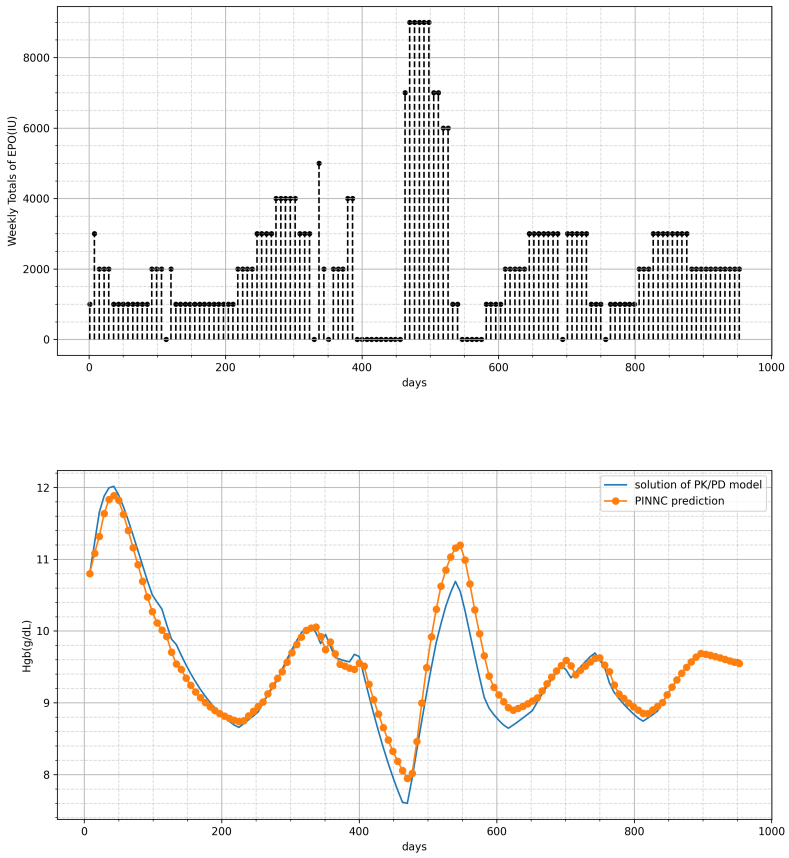


Figure 3.11: PINNC prediction based on another patient's EPO dosages

tion model. This framework makes PINN suitable for controller design. Meanwhile, this kind of neural network can also use physics information to improve performance. In the self-loop example of Van der Pol oscillator and Hgb response prediction, we have shown this PINNC model is accurate enough for control purposes.

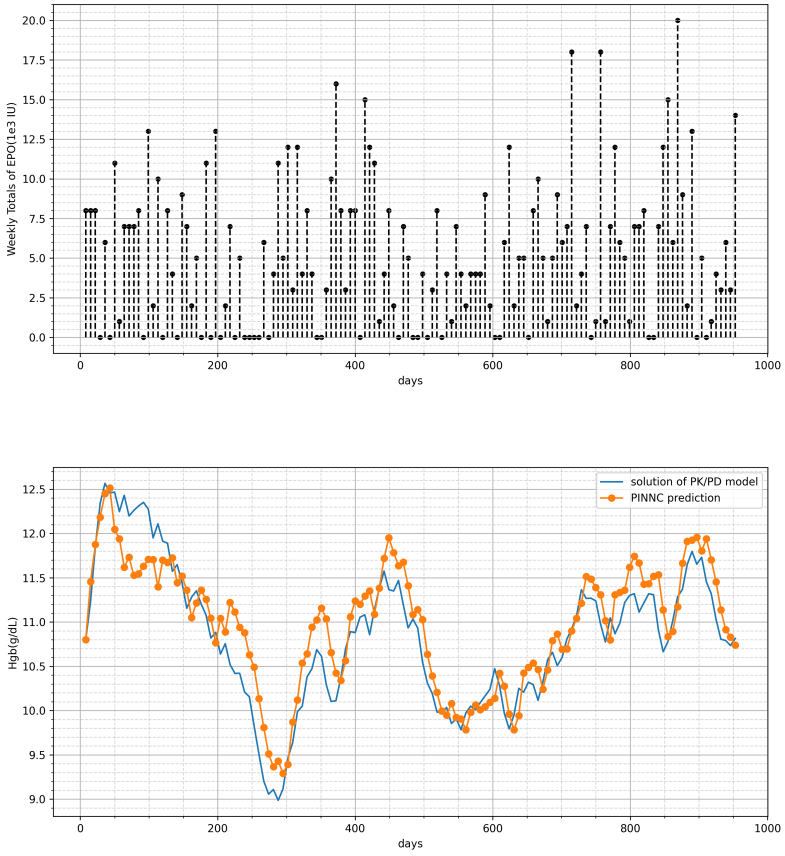


Figure 3.12: PINNC prediction based on random EPO dosages

Chapter 4

Model Predictive Control and Simulation Results

4.1 Introduction

In this chapter, the trained PINNC model described in chapter 3 will be applied to the MPC design. Since MPC was introduced in the 1970s, it has been successfully applied in the electric power technology[46], thermal energy storage systems[47] and process industries[48], as well as in robotics[49]. The main advantages of MPC over other controllers are its ability to handle system constraints, non-minimum phase processes, changes in models and its straightforward applicability to large, multi-variable processes.[50][51] In the following sections, we start with an introduction to MPC and then explore the control effect on Van der Pol oscillator and anemia treatment. Besides, the measurement noise and abnormal situations during treatment are considered.

4.2 Model Predictive Control

MPC development can be traced back to late 1970s when Richalet introduced the first MPC strategy based on quadratic programming.[52] The core idea of MPC is rolling-horizon optimization as shown in Figure 4.1. After sampling current states, MPC predicts future states based on prediction model. Then it will evaluate the predicted states and generate a series of optimal control inputs according to control

objectives on a finite prediction horizon. The first control action in the series will be implemented into system. After that, MPC goes to the next cycle and repeats the above steps.

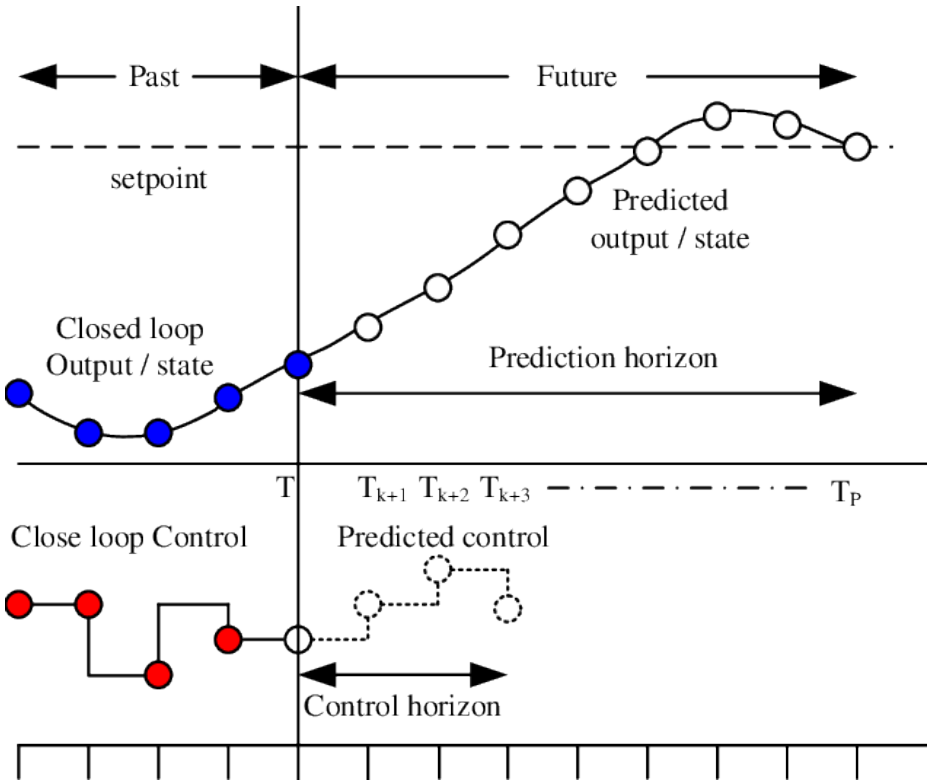


Figure 4.1: General concept for MPC[53]

The Block diagram of MPC system with PINNC is revealed in Figure4.2. The PINNC is connected to the plant system which is simulated by physiologically based differential equations in this experiment.

Usually, quadratic functions can be chosen as cost functions to penalize the error in the reference tracking. A general quadratic cost function for MPC optimization is

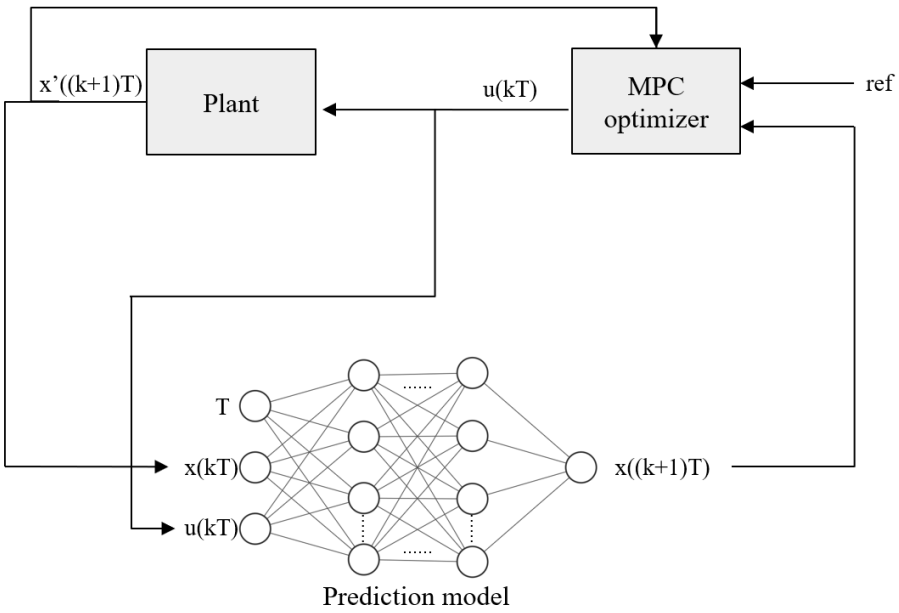


Figure 4.2: MPC structure with PINNC model

given as:

$$\min_{u[k],...,u[k+N-1]} J = \sum_{j=1}^{N} \|x[k+j] - x_{ref}[k+j]\|_Q^2 + \sum_{j=0}^{N-1} \|\Delta u[k+j]\|_R^2 \tag{4.1}$$

$$s.t.\ x[k+j+1] = f(x[k+j], u[k+j]),\ j = 0, ..., N-1 \tag{4.2}$$

$$\Delta u[k+j] = u[k+j] - u[k-1],\ j = 0, ..., N-1 \tag{4.3}$$

$$h(x[k+j], u[k+j]) \leq 0,\ j = 1, ..., N \tag{4.4}$$

$$g(x[k+j], u[k+j]) = 0,\ j = 1, ..., N \tag{4.5}$$

where k and $x[k]$ represent the recurrent time step and states of the dynamic system, x^{ref} is the set-point reference value over the prediction horizon, the penalization of the quadratic error between the model prediction x and the reference x^{ref} and control input changes Δu over the prediction horizon form the cost function J, matrices Q and R are weighting coefficients respectively, N is control horizon; Equation 4.2 is the constraint imposed by prediction model f, Equation 4.3 refers to the relation between the manipulated variable u and the control input changes, Equation 4.4 and Equation 4.5 represent the constraints in the real system like input range.[42]

Equations 4.1-4.5 define an optimization problem. There are many well-established methods. In this work, we use interior-point method based optimization solver.[54]

From the above discussion, we can see the classical MPC aims at a set-point target. It is fine if we want the system states to follow the desired trajectories. But for many practical control questions, the controller goal is to keep the desired state in a defined zone instead of a specific value. This target zone is often required in biomedical control systems like the regulation of blood glucose and the treatment of anemia. To integrate this target zone into MPC, we use zone model predictive control (zone-MPC)[55] for renal anemia treatment control. As shown in Figure 4.3, zone-MPC divides the trajectory into three parts including one permitted range and two undesired areas. Hgb predictions beyond the upper bound and below the lower bound are marked as yellow dots and red dots. The control target is to optimize predicted Hgb to stay in the permitted range, which is marked as green dots, through manipulating EPO dosages.

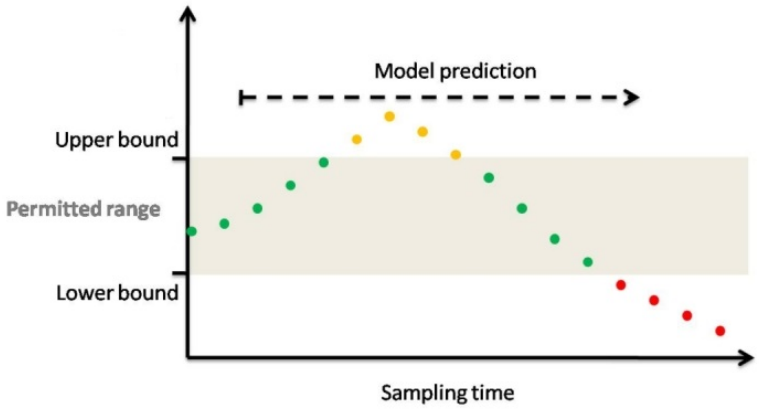


Figure 4.3: Zone model predictive control[55]

Like ordinary MPC, zone-MPC uses rolling optimization to calculate optimal manipulated variables and takes the first control input into action based on recurrent states. The significant part of zone-MPC is the cost function. Instead of driving the model output to a specific set point, the cost function of zone-MPC will penalize the prediction out of the defined zone and allow the model output to stay in the permitted

range. The cost function is illustrated below:

$$\min_{u[k],...,u[k+N-1]} J = \sum_{j=1}^{N} \|x_{range}[k+j])\|_Q^2 + \sum_{j=0}^{N-1} \|\Delta u[k+j]\|_R^2 \tag{4.6}$$

$$s.t.\ x_{range}[k+j] = C[k+j] \cdot (x[k+j] - x_{ref}[k+j]), j = 1, ..., N \tag{4.7}$$

$$C[k+j] = \begin{cases} 0,\ if\ x_{LB} \leq x[k+j] \leq x_{UB} \\ 1,\ otherwise \end{cases} \quad j = 1, ..., N \tag{4.8}$$

$$x_{ref}[k+j] = \frac{1}{2}(x_{LB} + x_{UB}),\ j = 0, ..., N-1 \tag{4.9}$$

$$x[k+j+1] = f(x[k+j], u[k+j]),\ j = 0, ..., N-1 \tag{4.10}$$

$$\Delta u[k+j] = u[k+j] - u[k-1],\ j = 0, ..., N-1 \tag{4.11}$$

$$h(x[k+j], u[k+j]) \leq 0,\ j = 1, ..., N \tag{4.12}$$

$$g(x[k+j], u[k+j]) = 0,\ j = 1, ..., N \tag{4.13}$$

$C[k+j]$ is a penalty coefficient, which penalizes the prediction out of the permitted range. There are multiple approaches to set the value of $x^{ref}[k+j]$. For example, we can choose the value of the upper bound and lower bound. In this work, the mean value of two bounds is the reference value in Equation 4.9. Actually, the effects of these methods are similar because finally, they can lead to similar cost function values by setting different weighting matrix Q. It has been proved that zone-MPC has robustness against plant model mismatch as well as against measurement noise.[56] But because there is no penalty to restrain states in the permitted range, the optimal output may be close to the upper bound or lower bound. To solve this problem, usually we can choose a tighter range than the actually desired scope.

4.3 Simulation Results

4.3.1 Van der Pol Oscillator Control

We still use the Van der Pol oscillator system as an example. The plant system is simulated by Equations 3.9 to 3.10. Prediction model is the PINNC model introduced

in chapter 3. Prediction horizon is 5T (2.5s). Control input value is limited in [-1, 1]. The first control target is to make state x_1 track objective trajectory using classical MPC. Diagonal matrix Q and R in cost function 4.1 are chosen as

$$Q = \begin{bmatrix} 20 & 0 \\ 0 & 0 \end{bmatrix} \quad R = [1] \tag{4.14}$$

The tracking result of 60 seconds is presented in Figure 4.4. Controlled state x_1 can fit the green dot reference trajectory very well.

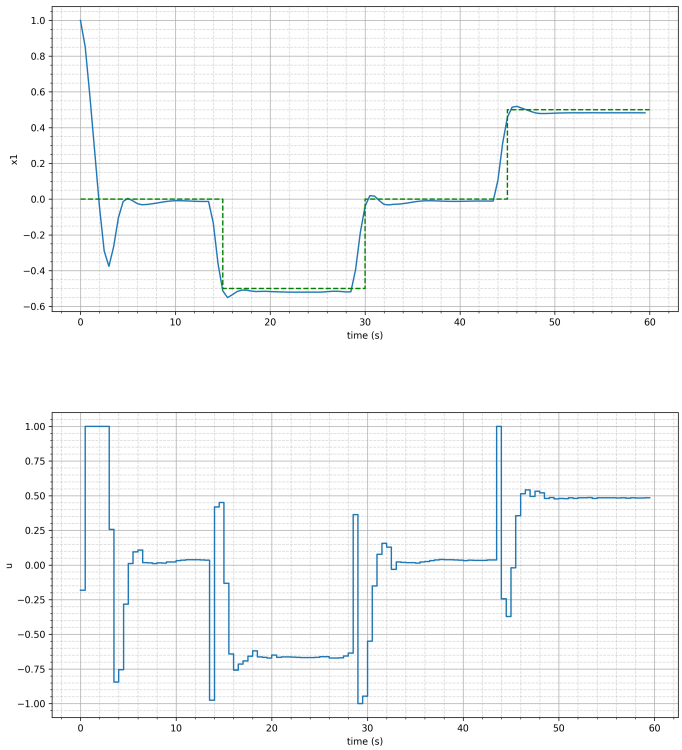


Figure 4.4: Classical MPC of the Van der Pol oscillator with PINNC

After introducing the classical MPC application in Van der Pol oscillator control, we explore further the performance of zone-MPC. The upper bound and lower bound of state x_1 in Equations 4.8, 4.9 are set as 0.5 and -0.5. The simulated result in Figure 4.5 meets our expectation. The x_1 trajectory oscillates near the bound.

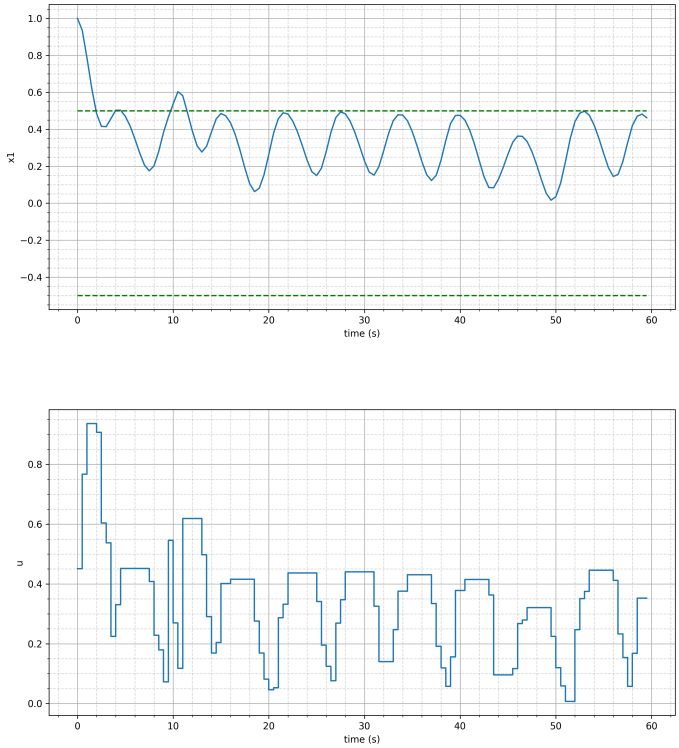


Figure 4.5: Zone-MPC of the Van der Pol oscillator with PINNC

4.3.2 Anemia Treatment Simulation Results

Finally, we implement the PINNC approximation model for automatic EPO dosages control in renal anemia treatment. The zone-MPC for this question is formulated by

$$\min_{EPO[k],...,EPO[k+N-1]} J = \sum_{j=1}^{N} \|R_{range}[k+j])\|_Q^2 + \sum_{j=0}^{N-1} \|\Delta EPO[k+j]\|_R^2 \quad (4.15)$$

$$s.t.\ R_{range}[k+j] = C[k+j] \cdot (R[k+j] - R_{ref}[k+j]), j = 1, ..., N \quad (4.16)$$

$$C[k+j] = \begin{cases} 0,\ if\ Hgb_{LB} \leq K_H \cdot R[k+j] \leq Hgb_{UB} \\ 1,\ otherwise \end{cases} \quad j = 1, ..., N \quad (4.17)$$

$$R_{ref}[k+j] = \frac{1}{2K_H}(Hgb_{LB} + Hgb_{UB}),\ j = 0, ..., N-1 \quad (4.18)$$

$$X[k+j+1] = f(T, X[k+j], EPO[k+j], EPO[k+j-1]),$$

$$X[k+j] = \{R[k+j], x_1[k+j], x_2[k+j]\},\ j = 0, ..., N-1 \quad (4.19)$$

$$\Delta EPO[k+j] = EPO[k+j] - EPO[k-1],\ j = 0, ..., N-1 \quad (4.20)$$

$$0 \leq EPO[k+j] \leq EPO_{max},\ j = 1, ..., N \quad (4.21)$$

In clinical treatment, the EPO dosage value is larger than zero. And in consideration of the patient's safety, we set an upper limit of EPO input which is the maximum EPO value in the patient's treatment records. This constraint is referred to as inequation 4.21. For this experiment patient, the maximum EPO is 20000. Related research has revealed the healthy range for Hgb is from 11.6 to 15 grams per deciliter.[57] We choose a tighter range [12, 14] for the lower bound and upper bound in Equations 4.17, 4.18. Additionally, we have discussed the abnormal situations during treatment in the last chapter. Internal bleeding and infections are simulated as well. The RBC population is multiplied by a parameter A_d as Equations 4.22,4.23.

$$R_{k,new} = A_d R_k \quad (4.22)$$

$$x_{2,k} = R_{k,new} - \frac{4x_{1,k}}{\mu} \quad (4.23)$$

This parameter is equal to 1 when there is no abnormal situation. Internal bleeding will cause an impulse decline of A_d, while an infection will lead to a step disturbance. This results in the descent of Hgb. These disturbances make this simulation more realistic and also test the model's robustness. The role of the plant system to be controlled is taken by the PK/PD Equations 2.1-2.8 with parameters in Table 3.1

which is solved by Euler method. The prediction horizon equals 4T (28 days). And the tuning parameters of zone-MPC are Q=1000, R=0.2. The result of 728 days' simulation can be seen in Figure 4.6. The patient's Hgb level starts at 8.5, but later it is basically stable at the healthy range. Although internal bleeding happens on the 175th day and infection takes place from 525 th day to 581 th day, the controller can successfully respond to these disturbances and drive the Hgb level back to normal.

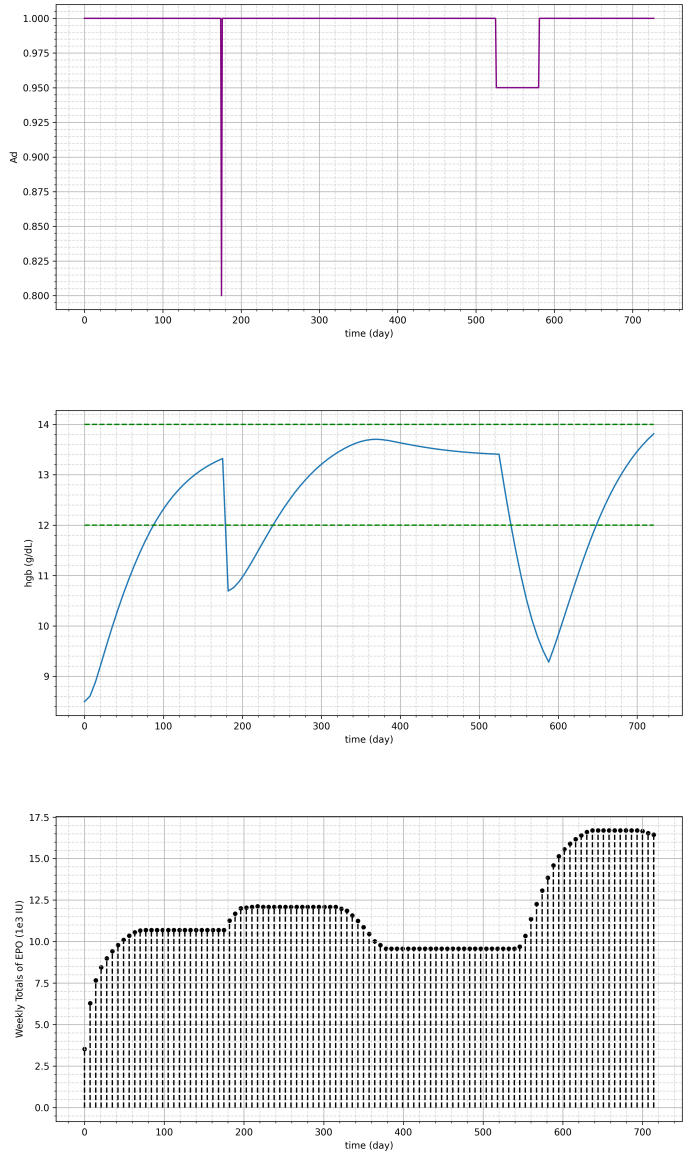


Figure 4.6: Solution of PINNC-based zone-MPC with disturbance

Furthermore, we take the measurement noise into consideration. It is modeled as a series of Gaussian noises which is added in the plant system and influences Hgb measurement value. The mean of the distribution is 0 while the standard deviation is 0.3. The disturbance and noise are combined together and generate the result in Figure 4.7. EPO dosages hold steady when Hgb is in the permitted range. If it is close to bound, the controller will change the dosage value. Generally, we still get a satisfactory control result lasting two years.

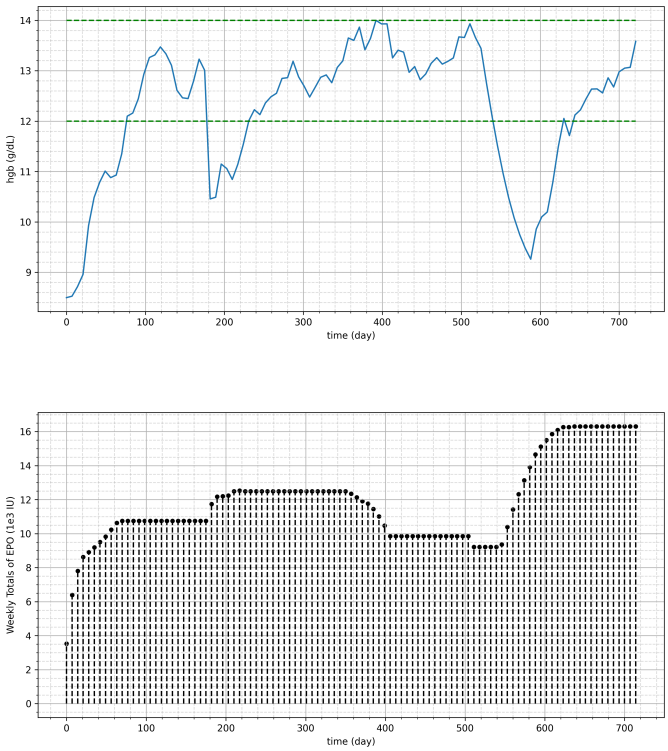


Figure 4.7: Solution of PINNC-based zone-MPC with disturbance and noise

4.4 Conclusions

In this chapter, we present the applicability of PINNC in the context of MPC for Van der Pol oscillator and anemia treatment. Benefitting from the accurate PINNC prediction model, MPC is able to find the optimal control input value. In the Van der Pol oscillator system, tracking trajectory and restricting the state within a target

zone is achieved through classical MPC and zone-MPC respectively. Facing the more complex anemia treatment system, zone-MPC can lead to the proper EPO dosage value. According to the optimized therapeutic schedule, the patient's Hgb level can be controlled in the target range. This zone-MPC system with PINNC model is qualified as a decision supporting tool.

Chapter 5

Conclusions and Future Work

In this thesis, we study the human body Hgb response modeling and control problem under EPO treatment. The main aim of the thesis is to investigate the physics-informed neural network based approaches, which have been proven to be an efficient method for combining the data-driven neural network modeling and the physical model obtained from first principles. The modified PK/PD model residuals are added to the loss function during training to force the neural networks to follow physical principles and infer unknown parameters. Since the original PINN is not created to handle changeable control actions and variable initial states, it is not suitable for control tasks. Therefore we propose to employ the improved PINNC with network input interfaces for control actions and initial states in MPC. The applications of PINNC in the context of MPC for Van der Pol oscillator and anemia treatment successfully solve the trajectory tracking and zone tracking problem. The optimal EPO dosage value can serve as a guide for clinical therapies.

In chapter 2, we apply PINN technique to model the Hgb response under EPO treatment. The differential equations with impulse input and the delay differential equations are replaced by the approximation equations and weighting functions. Benefiting from the incorporation of the physiological model into the neural networks, the test results show that the proposed method can build individualized models for patients with renal disease with good accuracy. But these proposed methods are

not general approaches and have some limitations. More complex differential equations will not be replaced easily. We are exploring the feasibility of using integration to convert the differential equations with impulse input so that the residual of the equation can be calculated. In addition, some papers have studied how to transform delay differential equations into ordinary differential equations or partial differential equations, which also inspires us to combine these methods with PINN in the future.

In chapter 3, to solve the problem that the original PINN is not capable of updating the initial states and control inputs, we trained the PINNC as the prediction model and it performs well on the self-loop prediction of Van der Pol oscillator and Hgb response example. Actually, this kind of neural network should also have the potential for parameter identification. We intend to use the PINNC for systems with uncertain parameters. Moreover, we mainly solve ordinary differential equations in this paper. In the future, we can apply this framework to differential-algebraic equations and partial differential equations as well.

Finally, during the controller design of chapter 4, the classical MPC and zone-MPC are chosen. Although they can achieve the desired effect, there are some improved methods like recursive zone MPC[58] and stochastic MPC using conditional value at risk[28]. We plan to try these methods which can bring better effects and solve more complicated questions.

Bibliography

[1] C. E. Lankhorst and J. B. Wish, "Anemia in renal disease: Diagnosis and management," *Blood reviews*, vol. 24, no. 1, pp. 39–47, 2010.

[2] A. Hayat, D. Haria, and M. O. Salifu, "Erythropoietin stimulating agents in the management of anemia of chronic kidney disease," *Patient preference and adherence*, vol. 2, p. 195, 2008.

[3] A. Guyton, *Textbook of medical physiology.: Saunders wb philadelphia*, 1986.

[4] B. D. Bradbury, M. D. Danese, M. Gleeson, and C. W. Critchlow, "Effect of epoetin alfa dose changes on hemoglobin and mortality in hemodialysis patients with hemoglobin levels persistently below 11 g/dl," *Clinical Journal of the American Society of Nephrology*, vol. 4, no. 3, pp. 630–637, 2009.

[5] S. Zendehboudi, N. Rezaei, and A. Lohi, "Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review," *Applied Energy*, vol. 228, pp. 2539–2566, Oct. 2018. DOI: 10.1016/j.apenergy.2018.06.051.

[6] L. Rajulapati, S. Chinta, B. Shyamala, and R. Rengaswamy, "Integration of machine learning and first principles models," *AIChE Journal*, vol. 68, no. 6, e17715, 2022.

[7] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine, "Enforcing analytic constraints in neural networks emulating physical systems," *Physical Review Letters*, vol. 126, no. 9, p. 98 302, Mar. 4, 2021. DOI: 10.1103/physrevlett.126.098302.

[8] Y. Jin and B. Sendhoff, "Knowledge incorporation into neural networks from fuzzy rules," *Neural Processing Letters*, vol. 10, no. 3, pp. 231–242, 1999. DOI: 10.1023/a:1018784510310.

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

[10] V. Bátora *et al.*, "The contribution of glucagon in an artificial pancreas for people with type 1 diabetes," in *2015 American Control Conference (ACC)*, IEEE, 2015, pp. 5097–5102.

[11] M. A. Nabian, R. J. Gladstone, and H. Meidani, "Efficient training of physics-informed neural networks via importance sampling," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 8, pp. 962–977, 2021.

[12] N. Thuerey, P. Holl, M. Mueller, P. Schnell, F. Trost, and K. Um, *Physics-based Deep Learning*. WWW, 2021. [Online]. Available: https://physicsbaseddeeplearning.org.

[13] D. E. Uehlinger, F. A. Gotch, and L. B. Sheiner, "A pharmacodynamic model of erythropoietin therapy for uremic anemia," *Clinical Pharmacology & Therapeutics*, vol. 51, no. 1, pp. 76–89, 1992.

[14] R. Ramakrishnan, W. K. Cheung, M. C. Wacholtz, N. Minton, and W. J. Jusko, "Pharmacokinetic and pharmacodynamic modeling of recombinant human erythropoietin after single and multiple doses in healthy volunteers," *The Journal of clinical pharmacology*, vol. 44, no. 9, pp. 991–1002, 2004.

[15] W. Krzyzanski, S. Woo, and W. J. Jusko, "Pharmacodynamic models for agents that alter production of natural cells with various distributions of lifespans," *Journal of pharmacokinetics and pharmacodynamics*, vol. 33, no. 2, pp. 125–166, 2006.

[16] P. Mendes and D. Kell, "Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation.," *Bioinformatics (Oxford, England)*, vol. 14, no. 10, pp. 869–883, 1998.

[17] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *computer*, vol. 27, no. 6, pp. 17–26, 1994.

[18] J. Waring, C. Lindvall, and R. Umeton, "Automated machine learning: Review of the state-of-the-art and opportunities for healthcare," *Artificial intelligence in medicine*, vol. 104, p. 101 822, 2020.

[19] A. Z. Woldaregay *et al.*, "Data-driven modeling and prediction of blood glucose dynamics: Machine learning applications in type 1 diabetes," *Artificial intelligence in medicine*, vol. 98, pp. 109–134, 2019.

[20] G. Schamberg, M. Badgeley, B. Meschede-Krasa, O. Kwon, and E. N. Brown, "Continuous action deep reinforcement learning for propofol dosing during general anesthesia," *Artificial Intelligence in Medicine*, vol. 123, p. 102 227, 2022.

[21] V Jahmunah *et al.*, "Automated detection of schizophrenia using nonlinear signal processing methods," *Artificial intelligence in medicine*, vol. 100, p. 101 698, 2019.

[22] A. E. Gaweda, A. A. Jacobs, M. E. Brier, and J. M. Zurada, "Pharmacodynamic population analysis in chronic renal failure using artificial neural networks—a comparative study," *Neural networks*, vol. 16, no. 5-6, pp. 841–845, 2003.

[23] A. E. Gaweda, M. K. Muezzinoglu, G. R. Aronoff, A. A. Jacobs, J. M. Zurada, and M. E. Brier, "Individualization of pharmacological anemia management using reinforcement learning," *Neural Networks*, vol. 18, no. 5-6, pp. 826–834, 2005.

[24] K. Shukla, A. D. Jagtap, J. L. Blackshire, D. Sparkman, and G. E. Karniadakis, "A physics-informed neural network for quantifying the microstructural properties of polycrystalline nickel using ultrasound data: A promising approach for solving inverse problems," *IEEE Signal Processing Magazine*, vol. 39, no. 1, pp. 68–77, 2021.

[25] Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 360, p. 112 789, 2020.

[26] A. Yazdani, L. Lu, M. Raissi, and G. E. Karniadakis, "Systems biology informed deep learning for inferring parameters and hidden dynamics," *PLoS computational biology*, vol. 16, no. 11, e1007575, 2020.

[27] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[28] J. McAllister, Z. Li, J. Liu, and U. Simonsmeier, "Epo dosage optimization for anemia management: Stochastic control under uncertainty using conditional value at risk," *Processes*, vol. 6, no. 5, p. 60, 2018.

[29] Y. Chait, J. Horowitz, B. Nichols, R. P. Shrestha, C. V. Hollot, and M. J. Germain, "Control-relevant erythropoiesis modeling in end-stage renal disease," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 3, pp. 658–664, 2013.

[30] J Duffus, "Glossary for chemists of terms used in toxicology (iupac recommendations 1993)," *Pure and applied chemistry*, vol. 65, no. 9, pp. 2003–2122, 1993.

[31] J. Ren, J. McAllister, Z. Li, J. Liu, and U. Simonsmeier, "Modeling of hemoglobin response to erythropoietin therapy through constrained optimization," in *2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, IEEE, 2017, pp. 245–250.

[32] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[33] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[34] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] T. Mitchell and G. Pegrum, "The oxygen affinity of haemoglobin in chronic renal failure," *British Journal of Haematology*, vol. 21, no. 4, pp. 463–472, 1971.

[37] R. Ando, H. Saito, and J. Takeuchi, "Factors that affect oxygen affinity of hemoglobin in chronic hemodialysis patients," *Nephron*, vol. 46, no. 3, pp. 268–272, 1987.

[38] J. V. Tembhurne and T. Diwan, "Sentiment analysis in textual, visual and multimodal inputs using recurrent neural networks," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 6871–6910, 2021.

[39] Y. Pan and J. Wang, "Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 8, pp. 3089–3101, 2011.

[40] P. Kittisupakorn, P. Thitiyasook, M. A. Hussain, and W. Daosud, "Neural network based model predictive control for a steel pickling process," *Journal of process control*, vol. 19, no. 4, pp. 579–590, 2009.

[41] S.-B. Yang, Z. Li, and J. Moreira, "A recurrent neural network-based approach for joint chance constrained stochastic optimal control," *Journal of Process Control*, vol. 116, pp. 209–220, 2022.

[42] E. A. Antonelo, E. Camponogara, L. O. Seman, E. R. de Souza, J. P. Jordanou, and J. F. Hubner, "Physics-informed neural nets for control of dynamical systems," *arXiv preprint arXiv:2104.02556*, 2021.

[43] J Mackenzei, "A. iserles a first course in the numerical analysis of differential equations (cambridge university press, cambridge, 1996), xvi+ 378pp., 0 521 55655 4 (paperback) 19.95 (us74. 95).," *Proceedings of the Edinburgh Mathematical Society*, vol. 41, no. 1, pp. 211–211, 1998.

[44] T. Kanamaru, "Van der pol oscillator," *Scholarpedia*, vol. 2, no. 1, p. 2202, 2007.

[45] S. J. Qin and T. A. Badgwell, "An overview of nonlinear model predictive control applications," *Nonlinear model predictive control*, pp. 369–392, 2000.

[46] J. Tobajas, F. Garcia-Torres, P. Roncero-Sánchez, J. Vázquez, L. Bellatreche, and E. Nieto, "Resilience-oriented schedule of microgrids with hybrid energy storage system using model predictive control," *Applied Energy*, vol. 306, p. 118 092, 2022.

[47] J. Tarragona, A. L. Pisello, C. Fernández, A. de Gracia, and L. F. Cabeza, "Systematic review on model predictive control strategies applied to active thermal energy storage systems," *Renewable and Sustainable Energy Reviews*, vol. 149, p. 111 385, 2021.

[48] M. A. Martins, A. E. Rodrigues, J. M. Loureiro, A. M. Ribeiro, and I. B. Nogueira, "Artificial intelligence-oriented economic non-linear model predictive control applied to a pressure swing adsorption unit: Syngas purification as a case study," *Separation and Purification Technology*, vol. 276, p. 119 333, 2021.

[49] M. Elsisi, K. Mahmoud, M. Lehtonen, and M. M. Darwish, "Effective nonlinear model predictive control scheme tuned by improved nn for robotic manipulators," *IEEE Access*, vol. 9, pp. 64 278–64 290, 2021.

[50] J Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.

[51] M Abu-Ayyad and R Dubay, "Real-time comparison of a number of predictive controllers," *ISA transactions*, vol. 46, no. 3, pp. 411–418, 2007.

[52] J. Richalet, A. Rault, J. Testud, and J Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[53] N Patcharaprakiti, J Thongpron, K Kirtikara, D Chenvidhya, and A Sangswang, "Model predictive control based on system identification of photovoltaic grid connected inverter," *International Journal of Information and Electronics Engineering*, vol. 2, no. 4, pp. 591–595, 2012.

[54] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[55] B. Grosman, E. Dassau, H. C. Zisser, L. Jovanovič, and F. J. Doyle III, "Zone model predictive control: A strategy to minimize hyper-and hypoglycemic events," *Journal of diabetes science and technology*, vol. 4, no. 4, pp. 961–975, 2010.

[56] R. Gondhalekar, E. Dassau, H. C. Zisser, and F. J. Doyle III, *Periodic-zone model predictive control for diurnal closed-loop operation of an artificial pancreas*, 2013.

[57] E. J. Benz, "Disorders of hemoglobin," *Harrison's principles of internal medicine*, vol. 18, pp. 852–61, 2008.

[58] J. McAllister, Z. Li, J. Liu, and U. Simonsmeier, "Erythropoietin dose optimization for anemia in chronic kidney disease using recursive zone model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1181–1193, 2018.