International Journal of Heat and Technology
Vol. 43, No. 1, February, 2025, pp. 241-250
Journal homepage: http://iieta.org/journals/ijht

Heat Management in Batch Reactors Using Flatness-Based Neural Predictive Control

Qasim Hamad[1,2*], Safanah Raafat[2]

[1] Electrical Engineering Department, University of Babylon, Babil 00964, Iraq
[2] Control and System Engineering Department, University of Technology, Baghdad 00964, Iraq

Corresponding Author Email: kasimalhussani@uobabylon.edu.iq

https://doi.org/10.18280/ijht.430125

Received: 15 December 2024
Revised: 28 January 2025
Accepted: 11 February 2025
Available online: 28 February 2025

Keywords:
differential flatness, model predictive control, neural network, model-free control, trajectory tracking, batch reactors

ABSTRACT

Effective heat management in batch reactors (BR) ensures accurate temperature control, enabling optimal reaction conditions, product quality, and process safety. Challenges in real-time temperature tracking involve balancing computational time with accuracy. Efficient Algorithms, as well as a simplified problem formulation, are essential in reducing computational complexity. This study exploits polymerization systems' differential flatness property (DFP) to simplify the temperature trajectory tracking problem. Expressing states and inputs as a function of flat outputs and their derivatives allows a lower-dimensional flat space, reducing computational complexity. A flatness-based model predictive control (FMPC) is proposed for efficient trajectory generation, enabling the handling of constraints while effectively producing feasible flat trajectories. This control architecture couples feedback from (MPC) with flatness feedforward linearization. It offers the computational advantage of requiring the solution of a convex quadratic programming (QP) instead of a nonlinear program. Meanwhile, a neural network-based model-free control (NN-MFC) has been designed to enhance the feedback loop of the FMPC and obtain improved trajectory tracking performance. In this paper, a nonlinear optimal tracking control problem for a nonlinear batch reactor is formulated and investigated. Moreover, a nonlinear model predictive control (NMPC) has been compared to the proposed control algorithm in crucial aspects such as tracking efficiency and computational complexity. The result of the proposed scheme has the advantages of excellent tracking and significantly reduced complexity compared to NMPC.

1. INTRODUCTION

In the industrial processes, the rapid progression led to increasing demands for automation and precision in complex systems. Accurate temperature control in batch reactors ensures dependable reaction kinetics, product quality, and process safety. Even minor temperature deviations can lead to undesired by-products, reduced yield, or even hazardous conditions. Given batch reactor dynamics' strong nonlinearities and time-varying nature, achieving precise temperature tracking remains a key challenge in industrial applications. Batch reactors are widely used in pharmaceuticals, polymer manufacturing, and specialty chemicals, where reaction conditions must be precisely controlled to ensure product quality and safety.

For example, in pharmaceutical synthesis, small temperature deviations can change reaction pathways, leading to the formation of undesired impurities that compromise drug efficacy. Similarly, maintaining an optimal temperature profile in polymerization processes is essential to control molecular weight distribution and polymer properties, directly impacting material performance. Beyond quality, temperature fluctuations can also pose significant safety risks, particularly in exothermic reactions, where an uncontrolled temperature rise may trigger runaway reactions or even thermal decomposition of reactants. This is a major concern in fine and petrochemical industries, where heat buildup can lead to reactor damage or hazardous conditions. Therefore, achieving precise and reliable temperature control is not just an academic challenge but a crucial requirement for industrial-scale batch processes.

Computationally efficient and accurate tracking of nonlinear systems is crucial due to the current advancement and sophisticated technologies that industries are adopting [1]. Traditional control methods are inadequate for real-time performances in the face of highly nonlinear systems [2]. An accurate and reliable system model is required to capture the inherent nonlinearities and enhance the control algorithms' effectiveness to optimize production in modern-day industries [3]. While using model-based control in linear setup offers simplicity and reasonable computational requirements, the latter falls short in handling highly nonlinear industrial systems. Meanwhile, leveraging accurate nonlinear models enables more advanced control strategies to ensure the design objectives [4]. Using a nonlinear model in an optimal control problem presents notable challenges and downfalls, such as computational and model complexity [5]. Consequently, various control strategies have been developed to address challenges related to computational complexity and the structural properties of the system.

Controlling chemical reactors primarily focuses on regulating process variables such as temperature, concentration, and pressure to ensure the quality of the product and operation safety [6]. The nonlinear, time-varying dynamics with complex reaction pathways of the reactors demand advanced trajectory tracking and optimization control strategies. In the literature, advanced control methods have been applied to deal with the issues of operating chemical reactors. Abdullah and Christofides [7] suggested integrating MPC and sparse identification of nonlinear dynamics SINDy, a recent identification technique to control and model nonlinear chemical process systems. In the study of Yadav et al. [8], data-driven modeling of a pilot plant BR was conducted, and the article used NMPC to track the temperature. Meanwhile, Obando et al. [9] used a dual-mode based sliding mode control SMC for temperature control for nonlinear chemical processes. The structural properties of chemical reactors were investigated by Nguyen et al. [10], a structural method based on the thermodynamics extended model that resulted in a control strategy to stabilize the dynamics of the reaction system. Linear algebra was used to design a controller based on the integral of desired closed-loop behavior in the study of Sardella et al. [11], with an application to regulation and trajectory tracking in a chemical process. Emphasizing the current state of advanced chemical and materials process control, Dubljevic [12] called for closing the large gap between developed novel process control theories and their industrial applications.

Model predictive control (MPC) is a powerful strategy in process control applications, providing significant rewards over traditional control methods such as PID controllers [13]. It was observed by Samad et al. [14] that MPC is more impactful than other control technologies, such as nonlinear and robust control, with a future impact of 85%. MPC utilizes an explicit process model to optimize control actions and predict future responses, enabling it to deal with constraints effectively. In the process industry, MPC aligns with process control initiatives, facilitating real-time implementation and monitoring, which maintain process integrity according to reference [15]. While MPC offers excellent performance with its linear replica, it faces several challenges, such as the need for precise system models, difficulty selecting its weight factors, and computational complexity [16]. NMPC differs from MPC in a fundamental concept, which is the uncalled-for step of model linearization. It utilizes a nonlinear model of the system and performs discretization, making predictions and controlling decisions based on this approach. Nonlinear solvers are employed to find the optimal control moves to satisfy an objective function, according to Li et al. [17]. Despite the advantages of NMPC over linear MPC, the former faces significant drawbacks that have limited its applications to a great extent. One of the most significant challenges practitioners encounter when implementing NMPC in real-time is its high computational demand. Different approaches were used to aid NMPC in overcoming its computational complexity; a successive online linearization technique was suggested by Patne et al. [18]. Linear MPC is an available method for most process control problems. As an alternative method to NMPC, it requires linearization of the nonlinear model and, hence, solving a convex optimization problem. However, Linear MPC suffers from complications in its performance due to the linearization process.

Researchers have advocated combining feedback linearization with MPC to address nonlinear system challenges. However, a major drawback of this approach lies in canceling nonlinear terms, which introduces significant robustness issues, even when an uncertainty model is incorporated into the control scheme. Additionally, the integration of MPC with feedback linearization (FBL) struggles to manage known input time delays effectively [19]. Despite these limitations, the method benefits from requiring the optimization problem to be solved via quadratic programming (QP), a computationally efficient process.

To overcome the robustness challenges associated with FBL, Flatness-Based Feedforward Linearization (FFL) emerges as an alternative. This approach addresses parametric uncertainties that can cause pole-zero cancellation inaccuracies, offering improved control design robustness. In their latest publication [20], the authors stated that most oversimplified systems models are differentially flat. Moreover, the ability to track an open-loop trajectory was established using flatness. Flatness is a decisive property that can facilitate the design of a control law to derive a linear or nonlinear system to the required objectives. Recent publications have focused on using the flatness property in many fields of control applications, such as aerospace [21, 22], industrial, and processes [23, 24], robotics, and mechatronics [25, 26]. Flatness can be used to design a flatness-based control in successive loops for subsystems [27, 28], or with other control techniques depending on the objective of the required design. In differentially flat systems, all system variables can be expressed in terms of flat output and a finite number of its time derivatives. This parameterization leads to reforming the system structure to find a global linearization without solving nonlinear differential equations. Differentially flat systems are the ones whose states and inputs are parametrized by flat outputs and a finite number of their time derivatives. Differential flatness (DF) input-output representation can fully capture the system's dynamic behavior in reduced, decoupled behavior, simplifying the trajectory generation problem. The flatness-based approach separates the nonlinear model into linear dynamics in Brunovsky form and nonlinear transformation [29]. A system is differentially flat if the states and the input can be parametrized according to a flat output and a finite number of time derivatives. Using any differential equation, these derivatives must be independent and unrelated to each other [30]. This method transforms the nonlinear system into an equivalent linear representation within the flat space. As a result, the system's dynamics are reformulated, yielding a linearized system in the transformed domain. This linear system facilitates using linear MPC instead of NMPC and enjoys utilizing quadratic programs for optimal control and linear control theory. Accordingly, a computationally efficient technique for the numerical solution of the optimal control problem can be obtained. By merging tools from the nonlinear control theory, the dimension of the original control problem is lowered, and the computational time is significantly enhanced [31]. On the other hand, the nonlinear transformation resulting from flatness can be learned using machine learning to close the feedback loop of the control system successfully. Machine learning techniques provide enhanced trajectory tracking when facing unpredictable operational issues [32, 33]. Nevertheless, learning the entire system dynamics is computationally expensive, and researchers adopted these approaches in the absence of an accurate model of the systems. Moreover, when using a data-driven model of the systems, the neural network fully engages with the dynamic system, which increases the execution time, as in the study of Shettigar et al. [34].

Our proposed methodology combines MPC with flatness feedforward linearization to improve the batch reactor's robustness property and tracking performance. The conventional dynamic model remains a realistic representation of the batch reactor; meanwhile, learning all the dynamics from the start would be inefficient. A more realistic scenario is to enhance the existing model and train the NN to learn the inverse dynamics. Integrating neural networks (NNs) with control strategies, such as adaptive and predictive control, is well-established to address system uncertainties and imperfections. Once trained, the NN plays a critical role in refining inputs to the control strategy, thereby enhancing overall system performance and robustness, as in the study of Li et al. [35]. Using a standard feedforward neural network (FNN) architecture, where layers are fully connected, ensures convergence to the desired values.

In our approach, we propose an efficient controller for temperature control of the batch reactor and attain a reliable tracking performance. The contributions of the article can be listed as follows:

•A novel control architecture where FMPC is used to couple flatness feedforward linearization with MPC in a closed form to establish a reliable tracking performance for the temperature profile of the batch reactor's heating phase.

•A feedforward neural network (FNN) is incorporated to enhance the performance of the proposed control strategy by learning the inverse dynamics introduced by flatness-based feedforward linearization. Consequently, the combined FNN with flatness-based feedforward linearization and FMPC enhances robustness and significantly improves tracking performance.

The proposed strategy achieved efficient execution time, leveraging the DF property of the batch reactor. It enables a direct generation of optimal trajectories and significantly reduces the computational burdens typically associated with NMPC.

This paper is organized as follows: Section 2 presents the dynamic model of the batch reactor. Background knowledge in Section 3. Section 4 presents the formulation of the proposed control strategy with the enhancement via a neural network. Section 5 displays the simulation results validating the proposed approach.

2. SYSTEM DESCRIPTION

A Batch Reactor (BR) is identified as a closed structure used primarily in processes that involve chemical and biochemical reactions. Unlike the continuous flow reaction, this reaction occurs in a contained batch for one time, where all the reactants are loaded at the start of the process. The controlled conditions for the reaction include temperature, pressure, and agitation. Once the response completes its operation cycles, the batch is unloaded, and the reactor is prepared for the next cycle after cleaning. The BR under consideration in this study is adopted from study [36], where the system dynamics involve the kinetics of the reaction and the energy balance for the reactor and the cooling jacket. The system schematics and description are shown in Figure 1.

Polymerization is commonly conducted in a batch reactor to produce polymers during a controlled reaction between monomers, initiators, and other reactants. The reaction operates under controlled temperature conditions to maintain product quality and safety. The refraction mechanism consists of the decomposition of the initiator, usually thermal or chemical decomposition, to start the polymer chain reaction. Meanwhile, the monomers react to form growing polymer chains in which temperature and reaction concentrations are critical. Then, the polymerization terminates due to the disproportion of the growing chain. The polymerization reactions are usually exothermic and require exact temperature control to prevent thermal degradation or runaway reactions.

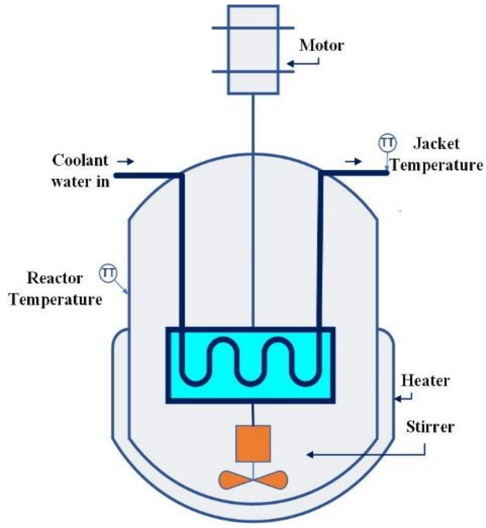

Figure 1. Batch reactor schematics

In this study, we consider a polymerization reaction with an initiator concentration. x_1 and monomer concentration x_2, and reactor temperature x_3 and jacket temperature as x_4 is considered. The BR system model is given by:

$$\dot{x} = f\big(x(t), u(t)\big)$$
$$= \begin{bmatrix} -A_d x_1 exp^{\left(\frac{-E_d}{Rx_3}\right)} \\ -A_p x_1 x_2 exp^{\left(\frac{-E_p}{Rx_3}\right)} \\ A_p V(\Delta H_p) x_1 x_2 exp^{\left(\frac{-E_p}{Rx_3}\right)} - UA(x_3 - x_4) + \frac{Q}{m_r c_{pr}} \\ UA(x_3 - x_4) - \frac{F_c c_{pc}(x_4 - T_c)}{m_j c_{pj}} \end{bmatrix} \quad (1)$$

Table 1. Batch reactor parameters [36]

Parameters	Value
A_d	4.4*1016 s-1
A_p	2.833*109 L/mol s
E_d	140.06*103 J
E_p	7.0711*104 J
R	8.3145 J/mol K
V	0.5 L
ΔH_p	-82.2*103 J/mol
T_c	27
$m_r c_{pr}$	5.9978*103 J/K
UA	27.0283 W/°C
Q	6.50*102 J/min
$m_j c_{pj}$	1.929*102
$x_1(0)$	1 mol/L
$x_2(0)$	1 mol/L
$x_3(0)$	45.30756°C
$x_4(0)$	45.30756°C
c_{pc}	4.184 J/g K

The polymerization batch reactor requires precise trajectory tracking for the temperature profile to ensure optimal performance. The system's highly nonlinear nature with complex reaction kinetics, coupled with the significant influence of the temperature on the process, makes reactor control challenging.

The flow rate is the manipulated variable, and it is subjected to constraints on its upper and lower limit, where the lower is zero, and the upper limit is 0.75 LPM. In Table 1, all the parameters used in the simulation are listed.

3. BACKGROUND KNOWLEDGE

3.1 Flatness

DF was introduced by M. Fliess and his colleagues and gained increasing attention in nonlinear control. Flatness is a natural idea associated with under-determined systems represented by differential equations. It signifies the possibility of representing the system's variables in terms of a finite set of free variables. A single input nonlinear system is said to be differentially flat if there exists a differential function of the state and a finite number of its time derivatives, called the flat output. As a structural property of systems, flatness allows for establishing the features required for designing feedback controller techniques, such as backstepping, passivity, and feedback linearization.

We consider a SISO system of the general form:

$$\dot{x} = f(x, u), x \in R^n, u \in R \tag{2}$$

where, $f = (f_1, \dots, f_n)$ is a smooth function of x and y and the rank of the Jacobian matrix is maximal.

A nonlinear system is considered differentially flat if there exist. $\zeta(t) \in R^m$, with differential independent components, and the following conditions are satisfied according to Fliess et al. [30]:

$$\zeta = \Lambda\left(x, u, \dots, u^{(\alpha)}\right) \tag{3}$$

$$x = \Phi(\zeta, \dot{\zeta}, \dots, \zeta^{(\rho-1)}) \tag{4}$$

$$u = \Psi^{-1}(\zeta, \dot{\zeta}, \dots, \zeta^{(\rho)}) \tag{5}$$

where, Λ, Φ and Ψ^{-1} are smooth functions in their domains, α and ρ are the maximum derivative order of the ζ and the control input u respectively. $\zeta = [\zeta_1, \dots, \zeta_m]^T$ is called flat output.

Systems that are classified as differentially flat can mathematically be represented by Brunovsky form, the states of which are called flat states:

$$z := [\zeta_1, \dot{\zeta_1}, \dots, \zeta_1^{(\rho_1-1)}, \dots, \zeta_m, \dots, \zeta_1^{(\rho_m-1)}] \tag{6}$$

where, ρ_i is the highest-order derivative of the flat output ζ_i as in Eq. (5). The state transformation between the flat state z and the system state x, which can be obtained by differentiating Eq. (3) and utilizing Eq. (4).

The nonlinear system can be put in the standard form:

$$\zeta_i^{(\rho_i)} = \alpha_i\left(\zeta, \dot{\zeta}, \dots, \zeta^{(\rho-1)}, u, \dot{u}, \dots, u^{(\sigma_i)}\right) := v_i \tag{7}$$

where, $\alpha_i, i = 1, \dots, m$ is a smooth function generated by the transformation. It's worth noting that σ_i represents the highest-order derivative of the control input u which yields after ρ_i times differentiation of the flat output ζ_i in Eq. (3).

The flat input is referred to as v and can be defined as:

$$v := [v_1, v_2, \dots, v_m]^T \tag{8}$$

Refreezing Eq. (7) and applying (6) and (8), we get:

$$\dot{z} = Az + Bv \tag{9}$$

$$v = \Psi\left(z, u, \dot{u}, \dots, u^{(\sigma)}\right) \tag{10}$$

where, $\sigma = \max \sigma_i$ and Eq. (9) is denoted as the linear flat model. The new definitions allow rephrasing Eq. (5) as follows:

$$u = \Psi^{-1}(z, v) \tag{11}$$

Up to this point, the fundamental difference between flatness feedforward linearization and feedback linearization can be explained by the following theorem:

Theorem 1 [37]: Consider a desired path in the flat space for the flat output ζ_d, with the corresponding desired flat state $\mathbf{z}_d$ and desired flat input v_d. Applying the nominal control $\boldsymbol{u} = \Psi^{-1}(\mathbf{z}_d, \mathrm{v}_d)$ to a differentially flat system, given ζ_d, and initial conditions as $\mathbf{z}(0) = \mathbf{z}_d(0)$, leads to a linear system (9) based on a change of coordinates. Noting that the variables $\mathbf{z}_d$, can be obtained from Eq. (6) by replacing the desired flat output and v_d are obtained from (7) by replacing the desired flat output.

Theorem 1 can enable controllers or trajectory generators to deal only with the linear flat model and produce the appropriate flat state and input as the output. Then, these outputs are fed through the inverse term (11) to offset the nonlinear term (10).

3.2 Model predictive control

Linear model predictive control (LMPC) is conveniently used with a linearized system as a control strategy for modifying the behavior of systems with linear approximation. LMPC is an optimization-based, predictive framework that belongs to the family of MPC. The critical aspect of LMPC is the use of a linear model in the prediction of the future behavior of the system over a finite prediction horizon. The optimization problem is dealt with at each time step to minimize a quadratic cost function subject to constraints.

Consider a discrete linear-time invariant system:

$$x_{k+1} = Ax_k + Bu_k \tag{12}$$

where, k is the discrete-time instant, $\mathbf{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state vector, $\mathbf{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input vector, A, B are appropriate size system matrices. $\mathbb{X}$, $\mathbb{U}$ are the state and control input constraint set, which are usually represented as linear inequality:

$$\begin{aligned} X &= \{x \in R^n : F_x \leq g_x\} \\ U &= \{u \in R^m : F_u \leq g_u\} \end{aligned} \tag{13}$$

The objective function is the balancing factor between tracking performance and the control input energy. It is responsible for minimizing the differences between the trajectories of the desired reference and states of the systems. Meanwhile, it penalizes the excessive control input signal using a quadratic cost function:

$$J_k = \sum_{k=0}^{N_p} x_k^T Q\, x_k + \sum_{k=0}^{N_c} u_k^T R\, u_k \tag{14}$$

where, Q and R are the weighting matrices, N_p, N_c are the prediction and control horizon, respectively.

Dealing with operational constraints is an essential feature of MPC, which can include limits on state variables and control inputs. The constraints can be incorporated directly into the formulation of the optimization problem to ensure the reliability of the operation under strict requirements.

QP is used to solve the optimization problem in LMPC due to the linear model and quadratic cost formulation. This feature is considered the main advantage of using LMPC due to the moderate computational efforts and the ability to solve in real time using standard optimization algorithms. This algorithm starts by describing the predicted state, which is a function of the current state and the input sequence:

$$X_k = A_k x_k + B_U U_k \tag{15}$$

The predicted state can be represented by the current state and control input. The cost function in Eq. (14) is subjected to the predictive state equation:

$$J_k = X_K^T Q_x X_k + U_k^T R_U U_k \tag{16}$$

Finally, by defining the appropriate-sized matrices to be incorporated into the prediction equations, the state constraints in Eq. (13) can be represented in terms of the current state and control input vectors as:

$$F_x X_k \leq g_x F_u U_k \leq g_u \tag{17}$$

A single decision vector that combines the state and control input vector can be formed to represent the new cost function:

$$\begin{gathered} \min_{\mathbf{z}} \mathbf{z}^T H\, \mathbf{z} \\ S.\text{ to: } Fz \leq g \\ F_{eq} z = g_{eq} \end{gathered} \tag{18}$$

Standard numerical optimization algorithms such as Newton's or the steepest-descent method can be used to solve QP problems. The optimization problem is solved at each time instant k and only the first element of the control vector is applied.

4. METHODOLOGY

This study's control method for the nonlinear batch reactor integrates flatness-based feedforward linearization with MPC, as illustrated in Figure 2.

The proposed method uses a linear flat model obtained by feedforward linearization in a feedback MPC to generate the desired flat states and input. These flat variables are then sent to the inverse term and applied to the nonlinear system. Mapping between the actual system state and the flat state is conducted, and the flat states are sent as feedback to MPC.

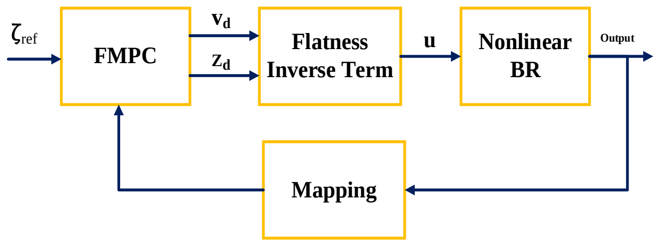

Figure 2. A schematic diagram of the proposed control system

4.1 Flatness formulation

The BR admits the flatness property; therefore, the nonlinear system (1) is flat by choosing the reactor temperature as the flat output and the input flow rate as the flat input. Thus, a flatness-based model predictive control (FMPC) structure is directly applied and achieves the desired temperature, which grants stability and optimality for the (BR). Define the flat output as:

$$\zeta = T_R \tag{19}$$

The flat state and the flat input are defined as:

$$z = [T_R, \dot{T}_R]^T \tag{20}$$

$$v = \ddot{T}_R \tag{21}$$

We can express the mapping between the flat state (4) and the actual BR states as:

$$z = \Phi^{-1}(x(t)) \tag{22}$$

The input $u(t)$ is mapped based on the flat state and the flat input as:

$$u(t) = \psi^{-1}(z, v) \tag{23}$$

Finally, introduce the following discrete linear flat system:

$$z_{k+1} = A_d z_k + B_d v_k \tag{24}$$

which represents a discrete-time linear system composed of a flat state and flat input.

4.2 Flatness-based model predictive control

An MPC controller based on flatness representation, as presented in Eq. (20) and Eq. (21), must be designed to implement the proposed control strategy. The prediction of the future flat states as a function of the flat input must be performed to predict the future flat output values and their derivatives. In the following flat state prediction equation and by using the matrix form, the prediction of the flat output and its derivatives is performed along the prediction horizon N_p:

$$z_{k,p} = \Gamma z_k + \Pi v_k \tag{25}$$

where,

$$\boldsymbol{z}_{k,p} = \begin{bmatrix} z_{k+1} \\ z_{k+2} \\ \vdots \\ z_{k+N_p} \end{bmatrix}, \boldsymbol{\Gamma} = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^{N_p} \end{bmatrix}, \boldsymbol{v}_k = \begin{bmatrix} v_k \\ v_{k+1} \\ \vdots \\ v_{k+N_P-1} \end{bmatrix}$$

$$\boldsymbol{\Pi} = \begin{bmatrix} B_d & 0_{nx.nu} \cdots 0_{nx.nu} & 0_{nx.nu} \\ A_d B_d & 0_{nx.nu} \cdots 0_{nx.nu} & 0_{nx.nu} \\ A_d^2 B_d & A_d B_d \cdots 0_{nx.nu} & 0_{nx.nu} \\ \vdots & \vdots \ \ddots \ \vdots & \vdots \\ A_d^{N_{p-1}} B_d & A_d^{N_{p-2}} B_d \ \cdots \ A_d B_d & B_d \end{bmatrix}$$

Up to this point, the prediction of the future flat states $z_{k,p}$ can be obtained since the future values of the flat output and its derivatives are available. Using state parametrization (22) and discretization, a conversion from the flat output and its derivatives to the original system states x_k can be applied at each sampling time.

The objective is to minimize the deviation between the actual system output and the reference trajectory, which can be incorporated into the new objective of the MPC controller.

The cost function is defined as:

$$\begin{gathered} \min_{v_k} \frac{1}{2} \sum_{k=1}^{N_p} (\zeta_k - \zeta_{k,ref})^T Q \left(\zeta_k - \zeta_{k,ref}\right) \\ + \frac{1}{2} \sum_{k=0}^{N_{p-1}} v_k^T R \, v_k \\ S.\ to{:}\ z_{k+1} = A z_k + B v_k \\ \zeta_k = C z_k \end{gathered} \tag{26}$$

This optimization problem can be solved using QP:

$$\begin{gathered} \min_{\chi} \frac{1}{2} \chi^T H \chi - G^T \chi \\ S.\ \text{to:}\ A x = b \end{gathered} \tag{27}$$

where, the decision vector is $\chi = [v_0, \ldots, v_{N_{p-1}}, z_0, \ldots, z_{N_p}]^T$ solving the optimization problem will result in determining the optimal input by substituting the optimal flat state and input in Eq. (24):

$$u_{op} = \Psi^{-1}(z_{op}, v_{op}) \tag{28}$$

The optimal control problem OCP in Eq. (26) is based on the utilization of a linear flat system obtained in Eq. (20) and Eq. (21). The appropriate choice of the flat output in (19), led to acquiring a linear system in the flat space that mimics the original nonlinear system in Eq. (1). The linear flat system is used in the OCP, MPC outputs z_{op} and v_{op} which are then fed through the inverse term (28).

In the FMPC OCP, the cost function J is subjected to a linear flat model, and a standard direct method strategy is considered. An open-loop OCP is solved at each sampling time by minimizing a quadratic cost function. This cost function is dependent on the sequence of predicted flat-state $\boldsymbol{z}_{k,p}$ and flat input $\boldsymbol{v}_k$ from (25). This procedure is subjected to a discretized linear flat model (24).

For this simulation, the prediction horizon N=10, and the control horizon=3. The weight matrix is *Q*=100, while *R*=0.01. Selecting the high value of *Q* is to ensure minimal deviation from the reference trajectory. Meanwhile, the value of *R* was adjusted to prevent aggressive actuator movements.

4.3 Learning-driven tracking

We propose using learning to enhance the accuracy of tracking a reference flat trajectory for FMPC. Using NNs to learn the inverse transformation of flatness feedforward linearization to compensate for any wrongful pole-zero cancelation. Inspired by the brain, NNs are computational models designed to identify patterns and relationships in data. They are formed by interconnected layers of neurons that process input data when entered through weighted connections, and with the help of activation functions, NNs can accurately approximate nonlinear mappings.

In the case presented in this paper, feedforward NNs, and FNNs were selected to enhance the feedback loop of FMPC, which represents the most common artificial neural network architecture. The structure of the FNNs is composed of an input layer, a hidden layer, which can be one or more, and finally, an output layer. Neurons are located in the hidden layer and represent the central processing unit in NNs. They receive the input, apply weight and activation functions, and produce outputs.

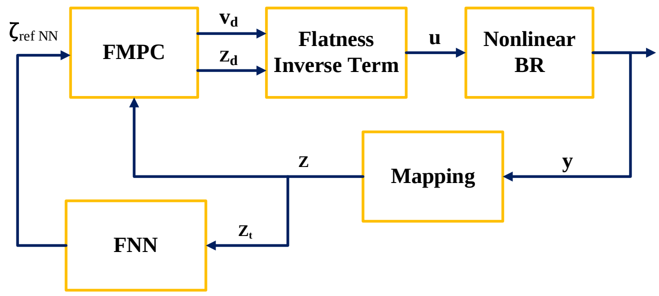

Figure 3. A schematic diagram of the proposed control system with neural network

To explain the architecture of the proposed FMPC-FNN and its underlying reasoning, we refer to Figure 3, which demonstrates the utilization of the FNN. The network feeds with the mapping output as its input and generates a modified flat trajectory as an output. The mapping can be obtained by the inverse function of Eq. (22), and the outcome is the system's original state that can be fed to the FNN.

The proposed scheme provides an accurate method to close the loop of the flatness feed-forward term with the MPC in the feedback loop. This method delivers an accurate trajectory that can be supplied to the FMPC to enhance the tracking performance for the proposed control structure.

In the following, a detailed explanation is provided of how the feedforward neural network (FNN) enhances the tracking performance of FMPC:

•FNN was used as a reference generator for the BR system. The network was trained to produce desired trajectories to be used as inputs to the FMPC scheme.

•A time vector served as an input to ensure that the generated desired trajectory is time-dependent. The actual output trajectory of the controlled system serves as the other input to the FNN.

•The output of the FNN generated a desired trajectory that was tailored as a reference path to guide the BR states.

•The training data consists of time and reference trajectory, paired with the corresponding desired output. The data was preprocessed as required to improve the FNN performance.

•The output of the FNN was utilized as the reference trajectory for FMPC. The role of FMPC is to minimize the deviation of the BR states from the FNN-generated reference trajectory while respecting system constraints.

5. SIMULATION RESULTS

This section presents the results of implementing the flatness-based model predictive control FMPC control strategy for the nonlinear BR system. A nonlinear MPC control strategy was also implemented to compare the results and assess the performance of FMPC. The results are accessed based on system performance, trajectory tracking precision, constrained control effort, and computational efficiency.

5.1 Temperature tracking response with FMPC

The presented figures provide valuable insights into the dynamics of the nonlinear BR under FMPC, highlighting its effectiveness in managing nonlinearity, achieving real-time feasibility, and ensuring system stability.

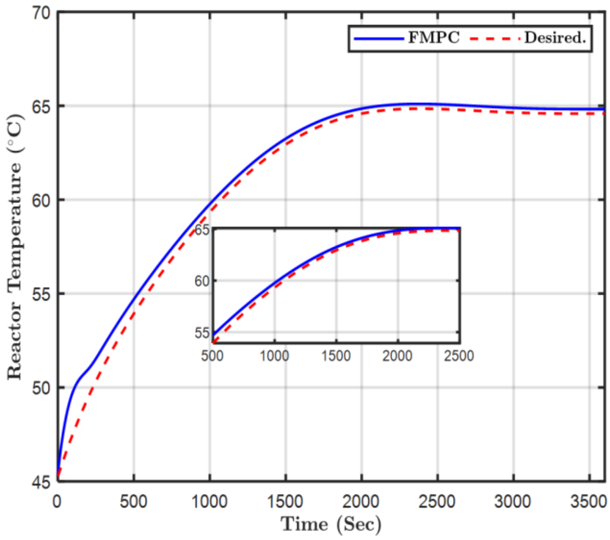

Figure 4. FMPC temperature tracking response

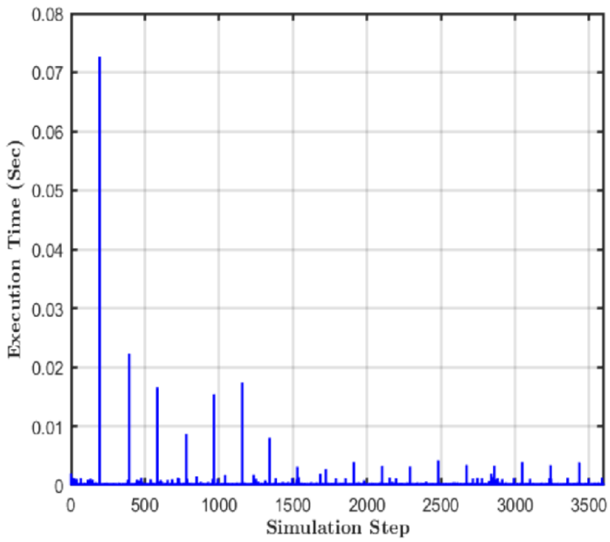

Figure 5. FMPC execution time analysis

Figure 4 illustrates the temperature tracking performance of the FMPC control strategy. The desired temperature trajectory is plotted together with the actual reactor temperature. Regarding tracking accuracy, FMPC demonstrates excellent tracking performance, which is evident by the close alignment between the actual and reference trajectories. Meanwhile, the controller ensured minimal overshoot and maintained stability.

FMPC execution time is depicted in Figure 5, a crucial metric in assessing the real-time feasibility of the controller. The relative reliability of execution time across simulation steps clearly indicates computational efficiency. This consistency is credited to the use of the flatness property of the BR, which simplifies the generation of feasible trajectories and reduces the computational complexity. The response has occasional peaks of 0.065 seconds at the highest, attributed to specific points where rapid changes in system dynamics occur. The average execution time is still way beyond the sampling time of 0.05, which was selected for the simulation of the BR, demonstrating the practicality of FMPC in real-time control applications.

5.2 Temperature tracking response with NMPC

The following figure illustrates the tracking performance of the NMPC applied to the nonlinear BR. Plotting the desired temperature trajectory with the actual temperature achieved utilizing FMPC.

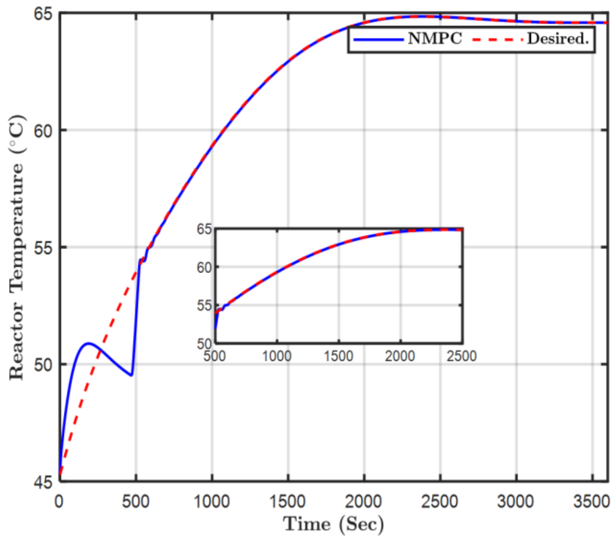

Figure 6. NMPC temperature tracking response

NMPC established high tracking accuracy when admitted to follow the nonlinear dynamics of BR, as shown in Figure 6. The discrepancies during the transient time are minimal due to NMPC's ability to manage changes in reference trajectory. The formulation of NMPC allows for explicitly handling BR system nonlinearity, reflected by the close alignment between actual and desired trajectories.

The response, with a peak execution time of 0.14, is almost three times the sampling period used in the simulation, which was selected as 0.5 seconds. Solving an optimization problem at each step period, NMPC ensures optimal control actions tailored to the BR nonlinear dynamics.

Although NMPC achieves excellent tracking, its computational demands can be significantly higher than FMPC. Figure 7 clearly demonstrates NMPC computational demands as its operation requires solving a nonlinear optimization problem. It is evident that the trade-off between tracking accuracy and computational efficiency is a critical consideration.

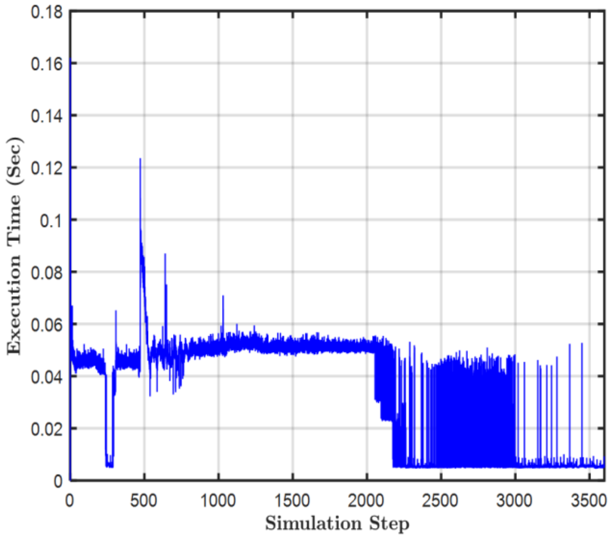

Figure 7. NMPC execution time analysis

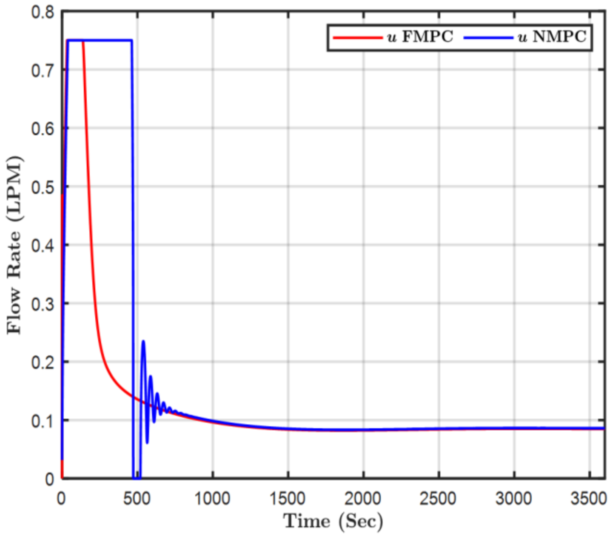

Figure 8. FMPC vs. NMPC constrained control signal comparison

The comparison between FMPC and NMPC-constrained control is illustrated in Figure 8. The constraints are imposed to ensure the physical limitations of the BR system are respected. The key observation is that FMPC and NMPC effectively maintain the control signal within the required boundaries. The FMPC control signal shows smoother variation than the NMPC control signal. NMPC exhibits more oscillatory behavior as compared to FMPC during transient time, which reflects the reactive nature of solving optimization problems dynamically. NMPC tends to respond more aggressively as the BR requires fast corrective action. Meanwhile, FMPC's smoother response prioritizes computational efficiency and stability.

5.3 Temperature tracking with NN-FMPC

The tracking performance after integrating the Feedforward Neural Network FNN is investigated in this section. Utilizing NN enhances FMPC by supplying an optimized or adaptive new reference trajectory as an input to FMPC. The key observation is the achievement of improved tracking accuracy compared to standard FMPC. The computational time has no significant impact as the NN is trained in advance and deployed efficiently. The execution time is still below the sampling period, which ensures real-time feasibility. Table 2 compares the three controllers in terms of their mean and max deviation.

Table 2. A comparison of control and temperature deviations

Parameter	Mean Control Deviation (LPM)	Max Control Deviation (LPM)	Max Temp. Deviation (℃)
FMPC	0.0001	0.1859	2.2445
NMPC	0.0003	0.3270	4.0371
NN-FMPC	0.0002	0.2930	2.2539

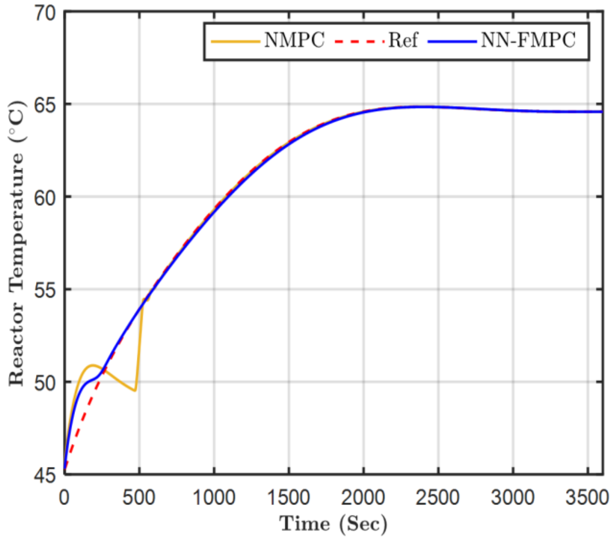

Figure 9. NN-FMPC vs. NMPC temperature tracking response

As depicted in Figure 9, the integration of NN into FMPC significantly reduces the gap in the tracking performance between FMPC and NMPC. When real-time implementation and energy efficiency are prioritized, NN-FMPC emerges as a superior option. A quantitative comparison between the three control approaches is presented in Table 3.

Table 3. A comparison between the three control approaches

Parameter	RMSE	ISE	IAE
FMPC	0.6317	1439	1683
NMPC	0.8133	2417	1035
NN-FMPC	0.3171	724.2	626
Parameter	**Ts (s)**	**Tr (s)**	**Mp%**
FMPC	236.5	822.1	0.8
NMPC	507.5	877.24	0.4
NN-FMPC	197.4	566.5	0.39

6. CONCLUSIONS

This study introduced a flatness-based model predictive control (FMPC) strategy for regulating the temperature of a polymerization batch reactor. The key innovation of FMPC lies in its ability to exploit the system's differential flatness property, enabling efficient trajectory generation and predictive control. Compared to nonlinear model predictive control (NMPC), FMPC demonstrated superior computational efficiency and reliable trajectory tracking, making it a promising candidate for real-time applications.

To further enhance FMPC's tracking precision and adaptability, this study proposed integrating a feedforward neural network (FNN) with FMPC. The FNN was trained to learn the inverse system dynamics, allowing it to refine the reference trajectory before being processed by FMPC. This NN-FMPC framework resulted in the lowest tracking errors (RMSE, ISE, and IAE), reduced settling time, and minimal overshoot, significantly outperforming both FMPC and NMPC. These findings highlight the advantage of combining data-driven learning with flatness-based control for improved robustness and efficiency.

The comparative analysis confirmed that NN-FMPC achieves the best trade-off between accuracy and computational effort, demonstrating its feasibility for real-time implementation. While FMPC alone provides fast computation and moderate tracking accuracy, NMPC exhibits higher computational costs and slower convergence.

This framework can be extended for future research to handle time-delayed systems, reaction kinetics variations, and reactant properties disturbances. Further optimization of FMPC, such as adaptive tuning of prediction horizons, could enhance performance under dynamic operating conditions. Finally, exploring different neural network architectures (e.g., recurrent networks or reinforcement learning-based models) could improve tracking accuracy and reduce execution time, reinforcing the potential of AI-driven enhancements in nonlinear control.

ACKNOWLEDGMENT

The authors extend their sincere gratitude to Prof. Dr. Thirunavukkarasu Indiran for generously providing the essential details and references related to the batch reactor system. His invaluable contribution has enhanced our understanding of the system's complexities and has been instrumental in facilitating the development of the controller.

REFERENCES

[1] Sa'id, W.K., Jassim, M.F., Raafat, S.M., Jasim, O.F. (2024). Induction furnace virtual temperature sensor based on robust adaptive estimation. In 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD), Erbil, Iraq, pp. 805-813. https://doi.org/10.1109/SSD61670.2024.10548813

[2] Le, D.M., Greene, M.L., Makumi, W.A., Dixon, W.E. (2021). Real-time modular deep neural network-based adaptive control of nonlinear systems. IEEE Control Systems Letters, 6: 476-481. https://doi.org/10.1109/LCSYS.2021.3081361

[3] Phan, D.T., Nguyen, L.M., Murali, P., Pham, N.H., Liu, H., Kalagnanam, J.R. (2021). Regression optimization for system-level production control. In 2021 American Control Conference (ACC), New Orleans, LA, USA, pp. 5023-5028. https://doi.org/10.23919/ACC50511.2021.9482638

[4] Hameed, A.H., Hameed, S.A., Humaidi, A.J. (2024). A comparative study of flux observers in induction motors. Iraqi Journal of Computers, Communications, Control & Systems Engineering, 24(1): 103-118. https://doi.org/10.33103/uot.ijccce.24.1.8

[5] Laghari, G.F., Malik, S.A., Daraz, A., Ullah, A., Alkhalifah, T., Aslam, S. (2022). A numerical approach for solving nonlinear optimal control problems using the hybrid scheme of fitness dependent optimizer and Bernstein polynomials. IEEE Access, 10: 50298-50313. https://doi.org/10.1109/ACCESS.2022.3173285

[6] Rominiyi, O.L., Akintunde, M.A., Bello, E.I., Lajide, L., Ikumapayi, O.M. (2023). Development and evaluation of a batch–reactor for catalytic depolymerization of polymeric waste for liquid and gaseous fuel production. International Journal of Heat and Technology, 41(6): 1596-1604. https://doi.org/10.18280/ijht.410622

[7] Abdullah, F., Christofides, P.D. (2023). Data-based modeling and control of nonlinear process systems using sparse identification: An overview of recent results. Computers & Chemical Engineering, 174: 108247. https://doi.org/10.1016/j.compchemeng.2023.108247

[8] Yadav, E.S., Shettigar J.P., Poojary, S., Chokkadi, S., Jeppu, G., Indiran, T. (2021). Data-driven modeling of a pilot plant batch reactor and validation of a nonlinear model predictive controller for dynamic temperature profile tracking. ACS Omega, 6(26): 16714-16721. https://doi.org/10.1021/acsomega.1c00087

[9] Obando, C., Rojas, R., Ulloa, F., Camacho, O. (2023). Dual-mode based sliding mode control approach for nonlinear chemical processes. ACS Omega, 8(10): 9511-9525. https://doi.org/10.1021/acsomega.2c08201

[10] Nguyen, T.S., Hoang, N.H., Tan, C.K., Hussain, M.A.B. (2022). Structural approach to the control of a reaction system using thermodynamics-based extent model. IFAC-PapersOnLine, 55(18): 21-26. https://doi.org/10.1016/j.ifacol.2022.08.024

[11] Sardella, M.F., Serrano, E., Camacho, O., Scaglia, G. (2020). Linear algebra controller design based on the integral of desired closed-loop behavior: Application to regulation and trajectory tracking in a typical chemical process. Industrial & Engineering Chemistry Research, 59(45): 20131-20140. https://doi.org/10.1021/acs.iecr.0c04799

[12] Dubljevic, S. (2022). Quo Vadis advanced chemical process control. The Canadian Journal of Chemical Engineering, 100(9): 2135-2139. https://doi.org/10.1002/cjce.24505

[13] Jones, J., Kindembe, D., Branton, H., Lawal, N., Montero, E.L., Mack, J., Shi, S., Patton, R., Montagu, G. (2023). Improved control strategies for the environment within cell culture bioreactors. Food and Bioproducts Processing, 138: 209-220. https://doi.org/10.1016/j.fbp.2023.02.004

[14] Samad, T., Bauer, M., Bortoff, S., Di Cairano, S., et al. (2020). Industry engagement with control research: Perspective and messages. Annual Reviews in Control, 49: 1-14. https://doi.org/10.1016/j.arcontrol.2020.03.002

[15] Eslami, T., Jungbauer, A. (2024). Control strategy for biopharmaceutical production by model predictive control. Biotechnology Progress, 40(2): e3426. https://doi.org/10.1002/btpr.3426

[16] Blaud, P.C., Chevrel, P., Claveau, F., Haurant, P., Mouraud, A. (2022). Four MPC implementations compared on the quadruple tank process benchmark: pros and cons of neural MPC. IFAC-PapersOnLine, 55(16): 344-349. https://doi.org/10.1016/j.ifacol.2022.09.048

[17] Li, W., Zhang, X., Wang, Y., Xie, S. (2024). Comparison of linear and nonlinear model predictive control in path

following of underactuated unmanned surface vehicles. Journal of Marine Science and Engineering, 12(4): 575. https://doi.org/10.3390/jmse12040575
[18] Patne, V., Ingole, D., Sonawane, D. (2022). Towards fast nonlinear model predictive control for embedded applications. IFAC-PapersOnLine, 55(22): 304-309. https://doi.org/10.1016/j.ifacol.2023.03.051
[19] Naimi, A., Deng, J., Vajpayee, V., Becerra, V., Shimjith, S.R., Arul, A.J. (2022). Nonlinear model predictive control using feedback linearization for a pressurized water nuclear power plant. IEEE Access, 10: 16544-16555. https://doi.org/10.1109/ACCESS.2022.3149790
[20] Join, C., Delaleau, E., Fliess, M. (2024). Flatness-based control revisited: The HEOL setting. Comptes Rendus. Mathématique, 362(G12): 1693-1706. https://doi.org/10.5802/crmath.674
[21] Kovryzhenko, Y., Li, N., Taheri, E. (2024). A control system design and implementation for autonomous quadrotors with real-time re-planning capability. Robotics, 13(9): 136. https://doi.org/10.3390/robotics13090136
[22] Alshahir, A., Kaaniche, K., Albekairi, M., Alshahr, S., Mekki, H., Sahbani, A., Alanazi, M.D. (2024). An advanced IBVS-flatness approach for real-time quadrotor navigation: A full control scheme in the image plane. Machines, 12(5): 350. https://doi.org/10.3390/machines12050350
[23] Hamad, Q.M., Raafat, S.M. (2024). A flatness-based trajectory tracking control for chemical reactor. In 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD): Erbil, Iraq, pp. 819-825. https://doi.org/10.1109/SSD61670.2024.10549742
[24] Hamad, Q., Raafat, S. (2024). Flatness-based model predictive constrained optimal control for chemical reactor. Journal Européen des Systèmes Automatisés, 57(6): 1583-1592. https://doi.org/10.18280/jesa.570605
[25] Abadi, A., Ayeb, A., Labbadi, M., Fofi, D., Bakir, T., Mekki, H. (2024). Robust tracking control of wheeled mobile robot based on differential flatness and sliding active disturbance rejection control: Simulations and experiments. Sensors, 24(9): 2849. https://doi.org/10.3390/s24092849
[26] Tonan, M., Bottin, M., Doria, A., Rosati, G. (2024). Motion planning of differentially flat planar underactuated robots. Robotics, 13(4): 57. https://doi.org/10.3390/robotics13040057
[27] Rigatos, G., Abbaszadeh, M., Pomares, J. (2023). Flatness-based control in successive loops for electropneumatic actuators and robots. IFAC Journal of Systems and Control, 25: 100222. https://doi.org/10.1016/j.ifacsc.2023.100222
[28] Rigatos, G., Busawon, K., Abbaszadeh, M., Pomares, J., Gao, Z., Cuccurullo, G. (2024). Flatness-based control in successive loops of the attitude of reentry space vehicles. In 2024 IEEE Conference on Control Technology and Applications (CCTA), Newcastle upon Tyne, United Kingdom, pp. 165-170. https://doi.org/10.1109/CCTA60707.2024.10666617
[29] Rigatos, G., Abbaszadeh, M., Pomares, J., Wira, P., Cuccurullo, G. (2024). Flatness-based control in successive loops for robotic manipulators and autonomous vehicles. International Journal of Systems Science, 55(5): 954-979. https://doi.org/10.1080/00207721.2023.2301040
[30] Fliess, M., Lévine, J., Martin, P., Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. International Journal of Control, 61(6): 1327-1361. https://doi.org/10.1080/00207179508921959
[31] Peherstorfer, B. (2022). Breaking the Kolmogorov barrier with nonlinear model reduction. Notices of the American Mathematical Society, 69(5): 725-733. https://doi.org/10.1090/noti2475
[32] Shin, Y., Smith, R., Hwang, S. (2020). Development of model predictive control system using an artificial neural network: A case study with a distillation column. Journal of Cleaner Production, 277: https://doi.org/124124.10.1016/j.jclepro.2020.124124
[33] Yel, E., Bezzo, N. (2021). A meta-learning-based trajectory tracking framework for UAVs under degraded conditions. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, pp. 6884-6890. https://doi.org/10.1109/IROS51168.2021.9635918
[34] Shettigar J.P., Kumbhare, J., Yadav, E.S., Indiran, T. (2022). Wiener-neural-network-based modeling and validation of generalized predictive control on a laboratory-scale batch reactor. ACS Omega, 7(19): 16341-16351. https://doi.org/10.1021/acsomega.1c07149
[35] Li, Y., Zhu, Q., Elahi, A. (2024). Quadcopter trajectory tracking control based on flatness model predictive control and neural network. Actuators, 13(4): 154. https://doi.org/10.3390/act13040154
[36] Prajwal Shettigar, J., Lochan, K., Jeppu, G., Palanki, S., Indiran, T. (2021). Development and validation of advanced nonlinear predictive control algorithms for trajectory tracking in batch polymerization. ACS Omega, 6(35): 22857-22865. https://doi.org/10.1021/acsomega.1c03386
[37] Hagenmeyer, V., Delaleau, E. (2003). Exact feedforward linearization based on differential flatness. International Journal of Control, 76(6): 537-556. https://doi.org/10.1080/0020717031000089570