IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 October 2023, accepted 28 November 2023, date of publication 5 December 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339571

RESEARCH ARTICLE

A Novel Adaptive LSSVR-Based Inverse Optimal Controller With Integrator for Nonlinear Non-Affine Systems

MUHAMMET EMRE SANCI[1], (Member, IEEE), KEMAL UÇAK[2], (Member, IEEE), AND GÜLAY ÖKE GÜNEL[1]

[1]Control and Automation Engineering Department, Istanbul Technical University, 34469 Istanbul, Turkey
[2]Electrical and Electronics Engineering Department, Mugla Sıtkı Kocman University, 48000 Mugla, Turkey

Corresponding author: Muhammet Emre Sanci (sanci17@itu.edu.tr)

ABSTRACT In this study, a novel inverse optimal controller based on NARMA-L2 modelling technique and online least squares support vector regression (LSSVR) method has been proposed for nonlinear non-affine systems. Firstly, the nonlinear autoregressive with exogenous inputs (NARX) model of the system is obtained using online LSSVR method, then this model is decomposed into NARMA-L2 submodels. Hence, the non-affine system model is converted to a nonlinear affine system model. The obtained NARMA-L2 submodels are used in computing the inverse optimal control law. Furthermore, the parameters of the inverse optimal controller have also been optimized online using the Levenberg-Marquadt algorithm. The performance of the proposed LSSVR based inverse optimal controller using NARMA-L2 model has been evaluated by simulations carried out on two benchmark systems, and the results show that the LSSVR based NARMA-L2 model and inverse optimal controller attain good modelling and control performances.

INDEX TERMS Adaptive control, inverse optimal control, NARMA-L2 model, online LSSVR.

I. INTRODUCTION

In model based adaptive control methods, performance depends on the estimation accuracy of the model. The system model must be accurately obtained to achieve good control results. For this purpose, machine learning based methods have been frequently used for precise modelling. Artificial neural networks (ANN) [1], [2], [3], [4], adaptive neuro fuzzy inference systems (ANFIS) [3], [4], [5] and support vector regressors (SVR) [6], [7], [8] have effectively been employed for model identification in model based control methods.

Due to the non-convexity of the objective functions of neural network-based methods trained with the backpropagation algorithm (ANN and ANFIS), these structures may become trapped at local minima, making it possible to obtain only a local model of the system.

In order to improve controller performance and minimize modelling inaccuracies, identification methods based on SVR have been recently proposed and applied due to their non-linear prediction and generalization competencies. In SVR, a convex objective function is formulated, hence gradient effects vanish and global extremum is ensured [9].

These machine learning based system identification methods generally start with an assumed model structure and then try to identify the input-output relation of the model. One of these presumed model structures is the so-called Nonlinear autoregressive moving average (NARMA)-L2 model. The NARMA-L2 model has been recently employed in various applications due to its practicality. In NARMA-L2 modelling, the Taylor series expansion of the nonlinear model is written down and first-order terms are retained. Hence, it is possible to have a model which is linear in input. In most of the works in technical literature NARMA-L2 models of systems are obtained offline [10], [11], [12], [13], [14], [15], [16], [17], [18].

From the viewpoint of control design, the computation of the control input is critical. In affine systems, the control input is separated from the system dynamics and appears linearly,

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng H. Zhu.

which makes it easier to calculate the control input. On the other hand, in non-affine system models, the control input is intermingled with the system dynamics and they cannot be separated. The study of non-affine nonlinear systems is an academically challenging subject, however the control input design for non-affine systems is a complicated and difficult task. The adaptive control literature contains numerous discussions and analyses of different kinds of nonlinear systems and nonlinear control methodologies [19]. Research on non-affine systems should not be disregarded since these types of systems have greater universality compared to affine systems. Most physical systems are inherently non-affine. At present, various control techniques have been developed for nonlinear non-affine systems, such as backstepping control [20], [21], [22], [23], adaptive control [24], [25], [26], optimal control [27], [28], [29], [30], [31], [32], [33], fuzzy control [34], [35], [36], [37], neural control [38], [39], [40], [41], [42], etc. As an alternative approach, in this paper, a novel least squares support vector regression (LSSVR) based inverse optimal controller with integrator is introduced for non-affine nonlinear dynamical systems.

The Inverse optimal control (IOC) method uses model information in computing the control law and therefore can be regarded as a model-based control method. The straightforward application of the optimal control theory leads to the Hamiltonian-Jacobi-Bellman (HJB) equation. For linear systems, the HJB equation can be easily solved yielding the linear regulator problem. For nonlinear problems, solving the HJB equation is not an easy task. Therefore, inverse optimal control methodology has been introduced to avoid the need to obtain the exact solution of the HJB equation for nonlinear systems. In literature, many studies have been conducted by using the inverse optimal control method. Numerical methods and global optimization techniques have been integrated with inverse optimal control to determine the optimal parameter values [43], [44], [45], [46].

The concept of inverse optimal control has been implemented in various fields of applications. Some examples of these from technical literature are summarized here. In [47], inverse optimal control has been applied to data obtained via motion capture of human-robot collaborative manipulation in a shared workspace. In [48], a basic predictive control structure has been designed for compensation of strict-feedforward nonlinear systems with input delays. Furthermore, it was shown that the proposed control is inverse optimal with respect to a meaningful differential game problem. Germinal Center Optimization algorithm (GCO) has been utilized to find an optimal set of parameters for a neural inverse optimal controller applied to an all-terrain tracked robot [49]. For inverse optimal control of multi-agent systems (MAS), first a consensus protocol is designed and then the cost function is optimized via Jaya algorithm (JA), teaching-learning algorithm (TLBO), and advanced teaching-learning (ATLBO) [45]. A neural affine system has been implemented to calculate the insulin delivery rate to control the glucose level in the blood for type 1 diabetes mellitus (T1DM) patients using inverse optimal control method [50]. In another study, an inverse optimal neural controller has been used for feedback control of mobile robots with non-holonomic constraints [51]. Based on the backstepping technique, neural networks have been utilized to approximate the unknown function, in the meantime, the inverse optimal controller has been used to minimize a meaningful cost function to reduce the cost of input [52].

A recurrent high order neural network (RHONN) has been used in order to identify the unknown system, then a control law which stabilizes the reference tracking error dynamics has been developed using the inverse optimal control approach [53]. RHONN structure has been utilized in various other applications of inverse optimal control. RHONN trained with extended Kalman filter (EKF) is employed to obtain a mathematical model for a three-phase linear induction motor (LIM) with uncertainties and discrete-time neural inverse optimal control is employed to control the position of the LIM [54]. Utilizing a RHONN which is trained with an EKF-based training algorithm, inverse optimal control technique is applied to a tracked robot [55]. In an attempt to apply control policies for personalized drug treatment in influenza infection disease, discrete-time impulsive systems are derived by combining inverse optimal control with a RHONN trained with EKF [56]. The passivity concept based inverse optimal controller integrated with RHONN has been utilized for stabilization of a bio-fuel production process [57]. RHONN was also utilized to identify the doubly fed induction generator and neural inverse optimal control was integrated for improvement of low-voltage ride-through capacity for a grid-connected doubly fed induction generator [58]. To achieve trajectory tracking for uncertain complex networks, a neural controller is applied to a small fraction of the nodes. The controller is composed of a RHONN and an inverse optimal controller to track the desired trajectory [59]. The combination of a RHONN identifier and neural inverse optimal control has been utilized to represent the viral dynamics of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [60].

An adaptive neural inverse optimal consensus control method is presented for MAS with uncertain dynamics and time-varying disturbance. Fuzzy logic (FL) has been utilized to identify the unknown nonlinear dynamics, and using a backstepping recursive design algorithm, an adaptive fuzzy inverse optimal scheme with a meaningful objective functional is developed [61]. In another application of FL, it has been utilized to identify the unknown nonlinear dynamics, hence firstly a fuzzy state observer is designed to estimate the immeasurable states. After that using the inverse optimal control principle and adaptive backstepping design theory, an observer-based fuzzy adaptive inverse optimal output feedback controller is developed [62]. In another work, FL has been employed to approximate unknown

nonlinearities and auxiliary system model of vehicular active suspension systems (ASS). Then combining adaptive backstepping design technique and inverse optimal principle, an adaptive fuzzy output feedback inverse optimal strategy is proposed [63].

In all of the examples given above from the technical literature, inverse optimal control methodology has been implemented for the control of affine nonlinear systems, where the control input can be separated from the nonlinear system dynamics in the mathematical model of the system. However, there is no significant research on the control of non-affine nonlinear systems using inverse optimal controllers. In this paper, a novel method has been proposed for the control of non-affine and nonlinear systems using inverse optimal control method. First, the nonlinear autoregressive network with exogenous inputs (NARX) model of the system to be controlled is obtained, consequently, this NARX model is converted into a NARMA-L2 model. Both the NARX and NARMA-L2 models have been identified by the online LSSVR method optimized by the Levenberg-Marquardt algorithm. The NARMA-L2 modelling provides a conversion from a non-affine to an affine system model. Next, using the computed affine system model, the inverse optimal controller is designed. The parameters of the controller have also been optimized with the Levenberg-Marquardt method. The resulting control architecture is an adaptive structure where the system model and control law are computed at every sampling time in an online adaptive manner.

Since the performance of model-based control methods depends on the model accuracy, updating the system model adaptively at every time step increases the success of the method. Furthermore, to the best of our knowledge, this is the first implementation of inverse optimal control methodology in literature in an adaptive manner, with optimized parameters at every sampling time.

Finally, we have also added an integrator controller to the design to decrease the steady-state error.

As mentioned, the main benefit of SVR-based approaches over backpropagation-based identification algorithms is that they guarantee the global extremum, allowing for the accurate determination of the system model that is valid in all regions. Thus, a novel online LSSVR-based inverse optimal controller has been proposed in this paper for nonlinear non-affine single input single output (SISO) dynamical systems, by combining the powerful modelling capabilities of LSSVR with the functionality of the inverse optimal controller structure.

The proposed work presents a number of significant contributions to the technical literature. These can be listed as:

1) A novel method is proposed to implement inverse optimal control methodology to nonlinear and non-affine systems. To the best of our knowledge, this is the first implementation of inverse optimal control technique to non-affine systems.

2) An online LSSVR has been employed with inverse optimal control method for the first time. Online LSSVR is used to convert the original non-affine system model (NARX) to affine system model using NARMA-L2 modelling technique.

3) Another major novelty of the proposed work is that the P and R parameters of the inverse optimal controller are made adaptive, they are updated iteratively using the Levenberg-Marquadt algorithm in an online manner.

4) Using inverse optimal control methodology and online LSSVR technique, an adaptive control architecture is implemented where model identification and control design are carried out iteratively, in an online manner.

The main contribution of our proposed method was to come up with a novel methodology to apply inverse optimal control technique to non-affine nonlinear systems. Nevertheless, there are numerous methods that have been used to control non-affine systems in the technical literature, however, our main motivation was to adopt specifically the inverse optimal control technique to control non-affine systems. The application of inverse optimal control method has been limited to affine nonlinear systems and it has not been formulated for non-affine systems, since the control input term is implicit in non-affine dynamical equations and it is generally not possible to find an explicit mathematical solution for it. Hence, the inverse optimal control law formulated for affine systems in technical literature cannot be used. So, in this paper, we propose a novel approach to overcome this limitation and apply inverse optimal control method for nonlinear and non-affine systems.

The proposed method has been tested by simulations on two different benchmark problems. The simulation results verify that the introduced control method can successfully provide good tracking control. Moreover, the robustness of the method is also justified by the simulation results obtained under disturbance, noise and parametric uncertainty.

This paper is organized as follows: Section II presents a concise introduction to nonlinear and non-affine systems. Section III gives a brief summary of the inverse optimal control method for trajectory tracking. Section IV explains the principles of NARMA-L2 modelling, additionally online LSSVR is summarized and the derivation of LSSVR submodels are given. In Section V, a detailed explanation of the proposed control architecture is presented. In Section VI, simulation results and performance analysis of the controller are given. In Section VII, a detailed comparison between the proposed control method and a conventional PID controller is presented. The paper ends with a brief conclusion in Section VIII.

II. NONLINEAR NON-AFFINE SYSTEMS

Optimal control of affine nonlinear systems is a well-studied problem in literature [64]. However, currently, there is no adequate work related to the optimal control of non-affine-in-control nonlinear systems. When it comes to solving the Riccati Equation (RE), Algebraic Riccati Equation

(ARE), and Hamilton-Jacobi-Bellman (HJB) equation for control systems, nonlinear non-affine systems are more complicated compared to linear or nonlinear affine-in-control systems [29]. The primary difficulty in the control problem of nonlinear non-affine systems is that the structure of the plant model does not conform to the conventional affine system format, as the control input u appears in a non-linear fashion. Hence, it is difficult to obtain an explicit solution for the control input.

The generic representation of a nonlinear and affine continuous system is:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where the vector of system states is represented as $x \in R^n$, control input signal vector is denoted by $u \in R^m$, $f(.)$ and $g(.)$ are nonlinear and smooth functions. In this representation, the control input is separated from the nonlinear dynamics and appears in linear form, this simplifies the design of the control law. Since the majority of physical systems are inherently nonlinear, they can not be modeled using the assumption that they are affine. Systems where the control input cannot be separated from the internal dynamics are called as non-affine systems. The most general formulation for nonlinear non-affine systems is given as follows:

$$\dot{x} = f(x, u) \tag{2}$$

When $f(x, u)$ is a smooth function, equation (2) can be rewritten in Taylor expansion form as in (3) [65]:

$$\dot{x} = f_0(x) + \sum_{j=1}^{k} f_j(x)u^{[j]} + R(x, u) \tag{3}$$

Hence, nonlinear and non-affine systems can be approximated with affine system models by retaining the first-order terms in (3). The design of control laws for non-linear non-affine discrete-time systems is a difficult task due to the control input nonlinearity or system output nonlinearity. Thus it is common to assume that the plant is affine, which means that the model is linear in control input. Nevertheless, many physical systems, for example, chemical reactions, PH neutralization processes, etc. are inherently nonlinear, In their models, control input cannot be separated from the nonlinear dynamics [65], [66], [67], [68], [69], [70].

In this work, a novel adaptive inverse optimal control methodology for nonlinear non-affine systems has been proposed. This method is based on derivation of the NARMA-L2 model of the system to be controlled and thereby converting from a non-affine system model to an affine system model. The main aim in this work is to explore the applicability of inverse optimal control method for nonlinear and non-affine systems. In the following section, some basic information about inverse optimal control method is provided.

III. INVERSE OPTIMAL CONTROL

It is currently a challenging problem to find effective approaches for optimal control of nonlinear non-affine systems. This is because the corresponding Hamilton-Jacobi-Bellman (HJB) equations are complicated and difficult to solve and have no feasible solutions when the systems dynamics are not explicit. As a result, there is a lack of sufficient research in this area [71], [72]. Thus in our study, we extended the boundaries of inverse optimal control to non-affine systems.

A. A BRIEF REVIEW OF INVERSE OPTIMAL CONTROL

The fundamental concept underlying inverse optimal control theory is to compute a feedback control law that achieves stabilization as the initial step. This control law is then employed in optimizing a meaningful cost function that takes into account both the state variables and the control inputs [73], [74], [75], [76], [77], [78]. This definition of inverse optimal control theory may sound perplexing when compared to the definition of the optimal control problem where the cost functional is predetermined to formulate a stabilizing control law. In inverse optimal control approach, a candidate control Lyapunov function (CLF) is used to construct an optimal control law directly without solving the HJB equation. A storage function is used as a CLF candidate and the inverse optimal control law is selected as an output feedback control, which is obtained as a result of solving the Bellman equation. Important concepts of inverse optimal control are given below [78], [79], [80].

Consider a nonlinear, discrete and affine system given as [81], [82], [83]:

$$\boldsymbol{x_{n+1}} = \boldsymbol{f(x_n)} + \boldsymbol{g(x_n)u_n} \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$, $u_n \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^m \to \mathbb{R}^{n\times m}$. Here, $\boldsymbol{x_n}$ denotes the state, u_n represents the control input, subscript n expresses the time index at time n, $n \in Z^+$ and $f(.)$, $g(.)$ are smooth functions where $f(0) = 0$ and $g(x_n) \neq 0$ for all $x_n \neq 0$.

The cost functional related to the trajectory of the system and the control input is constructed as given below:

$$\boldsymbol{C(z_n)} = \sum_{i=n}^{\infty} (l(\boldsymbol{z_i}) + \boldsymbol{u_i^T R u_i}) \tag{5}$$

Here we define the tracking error along the trajectory x_n as $z_n = x_n - x_{\delta,n}$ where $x_{\delta,n}$ denotes the desired trajectory of x_n. $C(z_n) : \mathbb{R}^n \to \mathbb{R}^+$, $l(z_n) : \mathbb{R}^n \to \mathbb{R}^+$ is a positive semi-definite function and $R : \mathbb{R}^n \to \mathbb{R}^{m\times m}$ represents a real symmetric and positive definite weighting matrix, here, superscripts denote matrix dimensions. The values in matrix R can either be chosen as constant or can depend on the system state to adjust the weighting on control efforts based on the state value. In the context of state feedback control design, it is assumed that the entire state x_n is accessible. The optimal value function $C(z_n)$ can be utilized as a Lyapunov

function and can be reformulated as follows:

$$\begin{aligned} V(z_n) &= (l(z_n) + u_n^T R u_n) \\ &\quad + \sum_{i=n+1}^{\infty} (l(z_i) + u_i^T R u_i) \\ &= (l(z_n) + u_n^T R u_n) + V(z_{n+1}) \end{aligned} \tag{6}$$

The boundary condition $V(0) = 0$ is necessary for V to be considered as a Lyapunov function. According to the Bellman optimality principle, in the case of infinite horizon optimization, the function $V(z_n)$ becomes time-invariant and satisfies the discrete-time (DT) Bellman equation.

$$V(z_n) = \min_{u_n}[(l(z_n) + u_n^T R u_n) + V(z_{n+1})] \tag{7}$$

$V(z_{n+1})$ depends on x_n, $x_{\delta,n}$ and u_n by taking one step ahead prediction for z_{n+1}. It is important to note that the DT Bellman equation is solved backward in time.

To acquire the necessary conditions that the optimal control law must satisfy, we define the discrete-time Hamiltonian $H(z_n, u_n)$ as:

$$H(z_n, u_n) = (l(z_n) + u_n^T R u_n) + V(z_{n+1}) - V(z_n) \tag{8}$$

The necessary condition for optimality of the control law is that $\frac{\partial H(u_n, z_n)}{\partial u_n} = 0$, hence:

$$0 = 2Ru_n + \frac{\partial V(z_{n+1})}{\partial u_n} \tag{9}$$

The chain rule for vectors is applied in (9) to get:

$$0 = 2Ru_n + g^T(x)\frac{\partial V(z_{n+1})}{\partial z_{n+1}} \tag{10}$$

When optimal control theory is implemented for trajectory tracking problem, the HJB equation is obtained as:

$$\begin{aligned} &l(z_n) + V(z_{n+1}) - V(z_n) \\ &+ \quad \frac{1}{4}\frac{\partial V^T(z_{n+1})}{\partial z_{n+1}} g^T(x_n) R^{-1}(z_n) g(x_n) \frac{\partial V(z_{n+1})}{\partial z_{n+1}} = 0 \end{aligned} \tag{11}$$

Solving (11) is a challenging task, therefore inverse optimal control technique has been proposed as an alternative method to overcome this difficulty.

Definition 1: Radially Unbounded Function A positive definite function $V(x_n)$ such that $V(x_n) \to \infty$ as $\|x_n\| \to \infty$ is called to be radially unbounded.

Definition 2: Control Lyapunov Function Assume that $V(x_n)$ is a radially unbounded function, with $V(x_n) > 0$, $\forall x_n \neq 0$ and $V(0) = 0$. If for any $x_n \in \mathbb{R}^n$ there exist real values u_n such that $\Delta V(x_n, u_n) < 0$, where the difference in Lyapunov function is defined as $\Delta V(x_n, u_n) = V(f(x_n) + g(x_n)u_n) - V(x_n)$, then $V(.)$ is called to be a discrete-time control Lyapunov function (CLF) for (4).

Theorem: Global Asymptotic Stability The equilibrium point $x_n = 0$ of equation (4) is globally asymptotically stable if there exists a function $V : \mathbb{R}^n \to \mathbb{R}$ such that the following conditions hold

(i) V is a positive definite, decrescent and radially unbounded function.

(ii) $-\Delta V(x_n, u_n)$ is a positive definite function, where $\Delta V(x_n, u_n) = V(x_{n+1}) - V(x_n)$.

Theorem [50]: The optimal control law to achieve trajectory tracking is formulated as:

$$u_n = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial V(z_{n+1})}{\partial z_{n+1}} \tag{12}$$

with the boundary condition $V(0) = 0$.

Equation (12) is an inverse optimal (globally) stabilizing control input along $x_{\delta,n}$ if the following two conditions are satisfied:

Condition 1) For the system represented by (4), equation (12) achieves (global) asymptotic stability of $x_n = 0$

Condition 2) The control Lyapunov function, $V(z_n)$ satisfies:

$$\overline{V} := V(z_{n+1}) - V(z_n) + u_{c_n}^T R(z_n) u_{c_n} \leq 0 \tag{13}$$

If $l(z_n) := -\overline{V}$, then $V(z_n)$ is a solution for (11) and it follows that the cost functional of the tracking error is minimized. To compute the inverse optimal control law for the trajectory tracking problem of nonlinear systems, first a CLF satisfying the necessary properties [50] is selected as:

$$V(z_n) = \frac{1}{2}z_n^T P z_n \qquad P = P^T \succ 0 \tag{14}$$

where the trajectory tracking error (z_n) is represented as:

$$z_n = x_n - x_{\delta,n} = \begin{bmatrix} (x_{1,n} - x_{1\delta,n}) \\ \vdots \\ (x_{N,n} - x_{N\delta,n}) \end{bmatrix} \tag{15}$$

The inverse optimal control law can be computed as:

$$\begin{aligned} u_n &= | -\frac{1}{4}R^{-1}g^T(x_n)\frac{\partial z_{n+1}^T P z_{n+1}}{\partial z_{n+1}} | \\ &= | -\frac{1}{2}(R + P_2(x_n))^{-1}P_1(x_n, x_{\delta,n}) | \end{aligned} \tag{16}$$

where

$$P_1(x_n, x_{\delta,n}) = \begin{cases} g^T(x_n)P(f(x_n) - x_{\delta n+1}) \\ for\, f(x_n) \succeq x_{\delta,n+1} \\ g^T(x_n)P(x_{\delta,n+1} - f(x_n)) \\ for\, f(x_n) \preceq x_{\delta,n+1} \end{cases} \tag{17}$$

and

$$P_2(x_n) = \frac{1}{2}g^T(x_n)Pg(x_n) \tag{18}$$

In (16), (17), (18), the P matrix must be selected as positive definite and symmetric. Equation (16) formulates the inverse optimal control law which also guarantees the minimization of the cost functional in (7). A detailed proof of the theorem given above is available in [50], [77], [79], and [84].

By the theorem given above, the inverse optimal control law for trajectory tracking problem can be calculated with the

selection of appropriate P and R matrices for discrete-time, affine-in-input, nonlinear system models.

It must be noted that the derivation of the inverse optimal control law summarized above is based on the selection of a positive definite Lyapunov function with a negative definite derivative. Hence, it is inherently stable as long as P and R matrices are appropriately selected.

The inverse optimal control technique summarized above has been applied to affine nonlinear systems in technical literature, however, we are not aware of any studies about its implementation for non-affine systems. In this paper, we propose a novel methodology where a non-affine nonlinear system is converted to an affine system using NARMA-L2 modelling technique and consequently inverse optimal control law can be computed.

B. INVERSE OPTIMAL CONTROL WITH INTEGRATOR

When we implemented the inverse optimal control methodology formulated above to the trajectory tracking problem of nonlinear and non-affine systems, we observed that the transient response was very good, however, there was a steady-state error. To minimize the steady state error an integral control action was added to the inverse optimal control law. Integral control is part of the well-known proportional-integral-derivative (PID) control and its main function is to decrease the observed steady-state error.

PID control is extensively employed in industry as the predominant control algorithm, and it has widespread adoption for industrial control purposes. PID control has gained widespread acceptance due to its robust performance in various operating conditions and its inherent simplicity which provides easy implementation. The PID algorithm, as its name implies, comprises three fundamental components: Proportional, integral, and derivative. So it utilizes the information on error, rate of change of error and integral of error in computing the control input. Consequently, there are three coefficients (or gains) to adjust the relative importance of these components and they are tuned to achieve an optimal response.

The proportional component of the PID algorithm is solely determined by the error between the reference signal and the measured output of the controlled system. Feeding back this error information will result in a decrease in the error, in accordance with the basic principles of feedback control theory. Typically, increasing the proportional gain results in a faster response of the control system. Nevertheless, if the proportional gain is excessively high, it can lead to oscillations in the process. If we increase the value of the proportional gain beyond a certain point, it will result in amplification in oscillations, leading to system instability where the oscillations may result in an unstable system beyond control.

The derivative part of the PID algorithm is directly proportional to the rate of change of the observed error between the reference signal and system output. Measuring the rate of change of error makes it possible to predict how fast the error is changing, hence the computed control signal can take an action before the error builds up. So, a decrease in the output is induced when the process variable exhibits a rapid increase. Raising the derivative gain will increase the responsiveness of the control system with respect to variations in the error term, thereby increasing the overall speed of the control system's response. The derivative component of the PID algorithm adds an anticipation property to the control signal.

The integral component of the PID algorithm sums the error term over time. Unless the error is zero, the integral response of the PID controller continuously tends to increase over time. As a result, it reduces or eliminates the steady-state error. The steady-state error is the difference between the system output and the reference signal after the effect of the transients are over. The mathematical justification of this is that adding integral control is equivalent to increasing system type by one, which means that the number of open-loop poles at the origin of the complex plane is increased. It can be mathematically verified that adding an open-loop pole to the origin decreases the steady-state error [85], [86].

The PID control law can be formulated in discrete-time as:

$$u_{pid_n} = K_p\left(e_n + \frac{T}{T_i}\sum_{n=1}^{N} e_n + \frac{T_d}{T}(e_n - e_{n-1})\right) \tag{19}$$

Here e_n is trajectory tracking error, T is the sampling interval, T_i is integration time, T_d is derivative time, and K_p is the proportional gain.

In this study, to minimize the observed steady-state error when only the inverse optimal controller is used, an integrator control action is added to the controller design. The main advantage of using an integral control term is to decrease the steady-state error as described in detail above.

Hence, the overall control law implemented in this work is given by the following equation:

$$u_n = | -\frac{1}{2}(R + P_2(x_n))^{-1}P_1(x_n, x_{\delta,n}) | + K_I\left[\sum_{i=1}^{N} z_i\right] \tag{20}$$

where $K_I \in Z^+$ is optimized by Levenberg-Marquadt algorithm at every sampling time.

IV. LSSVR BASED NARMA-L2 MODEL OF NONLINEAR NON-AFFINE SYSTEMS

The main aim of this study is to develop a novel method to design an inverse optimal controller for non-affine and nonlinear systems.

In order to achieve this, the NARX model of the system is acquired initially, and then using this NARX model, the $NARMA-L2$ model is attained. Thus at the first step NARX model of the system is obtained by the available input-output data easily.

The dynamics of the NARX model should be decomposed into a $NARMA-L2$ model in order to apply an inverse

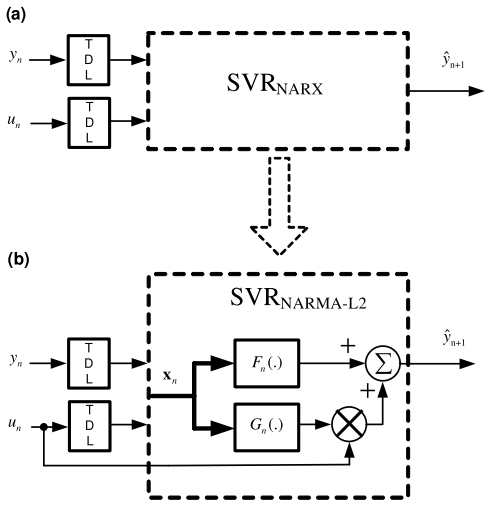


FIGURE 1. Decomposition of $LSSVR_{NARX}$ model to $LSSVR_{NARMA-L2}$ model [88]

optimal controller [87], [88], [89], [90]. Therefore, obtaining the $NARMA-L2$ model is a critical point.

The online LSSVR method has been used both to obtain the NARX model and then to convert it to the $NARMA-L2$ model.

Two separate LSSVRs are designed as presented in Fig.(1). $LSSVR_{NARX}$ computes the NARX model of the system from the input-output dataset of the system, then $LSSVR_{NARMA-L2}$ decomposes this model to a $NARMA-L2$ model in order to obtain a suitable system model to design the inverse optimal controller.

The proposed method employs online LSSVR method and NARMA-L2 modelling technique. Hence, in this section, some basic background material about these concepts are provided.

A. NONLINEAR FUNCTION ESTIMATION USING LSSVR

LSSVR is a regression technique which is based on the traditional SVR approach [91]. The main distinctive feature of LSSVR is that it utilizes equality constraints and a squared error term in the formulation of the optimization problem.

Let us assume that a training set is given as:

$$(y_1, \underline{x_1}), \cdots, (y_k, \underline{x_k}), x \in R^n, y \in R, k = 1, 2, \cdots, N \quad (21)$$

where N is the size of training data and superscript n is the dimension of the input matrix.The optimization objective in $LSSVR$ has two goals, it seeks to maximize the geometric margin while simultaneously minimizing the training error. The optimization problem is defined as:

$$\min_{w,b,e} \frac{1}{2} w^T w + \frac{1}{2} \sum_{k=1}^{N} e_k^2 \quad (22)$$

subject to equality constraints:

$$y_k - w^T \varphi(x_k) - b - e_k = 0, k = 1, 2, \cdots, N \quad (23)$$

The equations (22) and (23) are called as the primal form of the minimization problem where parameter w is weight vector, $\varphi(x_k)$ is nonlinear mapping which transfers the input dataset to feature space, b is bias, e_k error of least-squares data fitting. The fundamental concept behind support vector machines (SVMs) involves constructing a Lagrange function by combining the primal objective function with the associated constraints, which entails introducing a dual set of variables. The primal objective function and corresponding constraints are employed to obtain a dual set of variables and to construct the Lagrange function of the LSSVRs [92]. Thus, at first Lagrangian function can be derived by utilization of primal objective function and the constraints:

$$L(w, b, e, a) = \frac{1}{2} w^T w + \frac{1}{2} C \sum_{k=1}^{N} e_k^2 - \sum_{k=1}^{N} \alpha_k (w^T \varphi(x_k) + b + e_k - y_k) \quad (24)$$

where L represents the Lagrangian, C is the penalty term and a_k are Lagrange multipliers [8], [92], [93]. Since the Lagrangian exhibits a saddle point concerning both the primal and dual variables at the solution, the partial derivatives of L in relation to the primal variables must vanish to achieve optimality. Utilizing Karush-Kuhn-Tucker conditions for optimality yields:

$$\frac{\partial L}{\partial b} = \sum_{k=1}^{N} \alpha_k = 0 \quad (25)$$

$$\frac{\partial L}{\partial w} = w - \sum_{k=1}^{N} \alpha_k \phi(x_k) = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \quad (26)$$

$$\frac{\partial L}{\partial e_k} = C \sum_{k=1}^{N} e_k - \sum_{k=1}^{N} \alpha_k = 0 \rightarrow \alpha_k = C e_k \quad (27)$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \rightarrow y_k = w^T \varphi(x_k) + b + e_k \quad (28)$$

where $k = 1, 2, \cdots, N$. Thus, the solution is obtained as:

$$\begin{bmatrix} 0 & 1 \\ 1^T & \Omega + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ a^T \end{bmatrix} = \begin{bmatrix} 0 \\ y^T \end{bmatrix} \quad (29)$$

where $y = [y_1, y_2, \cdots y_N], a = [a_1, a_2 \cdots a_N], 1 = [1, 1 \cdots 1], \Omega = K(x_k, x_m), \quad k, m \in \{1, 2, \cdots N\}$, here K is kernel function.

The nonlinear autoregressive with exogenous inputs (NARX) model is given as:

$$y_n = f\left(u_n, .., u_{n-n_u}, y_{n-1}, .., y_{n-n_y}\right) \quad (30)$$

where u_n is the control input applied to the plant at time n, y_n is the output of the plant, and n_u and n_y denote the order of the NARX Model. This model can be employed to represent the dynamics of the non-linear system. The state vector of the system at time index n is represented as follows:

$$c_n = \left[u_n, .., u_{n-n_u}, y_{n-1}, .., y_{n-n_y}\right] \quad (31)$$

The estimate of the output $\hat{y}_n$ can be obtained using LSSVR. Consequently,using equations (23), (29) and (31), the non-linear regression or approximation function can be expressed in terms of support vectors as given in (32):

$$\hat{y}_n = \sum_{k \in SV} a_k K(c_n, \underline{x_k}) + b \tag{32}$$

For additional information please refer to [8], [92], and [93].

B. SYSTEM IDENTIFICATION USING NARMA-L2 MODEL

The nonlinear autoregressive moving average (NARMA) model is a discrete-time mathematical representation which is specifically designed for characterizing the behavior of a single input single output (SISO) system with nth-order nonlinear dynamics and a relative degree of d, in the neighborhood of its equilibrium state [94].

The generic representation of a discrete-time, non-affine and nonlinear system is given as:

$$y_{n+d} = F_{actual}(u_n, \cdots, u_{n-k+1}, y_n, \cdots, y_{n-k+1}) \tag{33}$$

Here y_n is the system output, u_n denotes the control input and d represents the relative degree, F_n is a nonlinear function of its arguments. For the system given above, solving system identification problem is equivalent to finding the correct input-output mapping with as small error as possible.

In literature, there are two variations of *NARMA*, known as $NARMA-L1$ and $NARMA-L2$. In both approximations, the relationship between the input u_n and the output y_n exhibits linearity. Consequently, the control law u_n can be determined by using algebraic linear equations. The key distinction between these two approximations lies in the Taylor series expansion which is used. For $NARMA-L1$, the Taylor expansion is centered around specific conditions: $[y_n, y_{n-1}, \cdots, y_{n-k+1}, u_n = 0, u_{n-1} = 0, \cdots, u_{n-k+1} = 0]$ while the Taylor expansion for $NARMA-L2$ is around the scalar $u_n = 0$. In the $NARMA-L1$ model, the functions $\hat{f}$ and $\hat{g}$ depend on previous values of the output, y_n, and the control input, u_n, with a linear relationship between the current and past control input values in the equation. However, in the $NARMA-L2$ model, the functions $\hat{f}$ and $\hat{g}$ involve both past output, y, and control input, u, with a linear connection between the next output value, y_{n+d}, and the current control input, u_n. The $NARMA-L1$ approximation model is given as:

$$\begin{aligned}\hat{y}_{n+d} =& \hat{f}[y_n, y_{n-1}, \cdots, y_{n-k+1}] \\ &+ \sum_{i=1}^{n-1} \hat{g}_i[y_n, y_{n-1}, \cdots, y_{n-k+1}]u_{n-i}\end{aligned} \tag{34}$$

where

$$\begin{aligned}\hat{f} &= F_{actual}[(y_n, \cdots, y_{n-k+1}, 0, 0, \cdots, 0)] \\ \hat{g}_i &= \left.\frac{\partial F_{actual}}{\partial u_{n-i}}\right|_{(y_n, \cdots, y_{n-k+1}; u_n=0, \cdots, u_{n-k+1}=0)}\end{aligned} \tag{35}$$

In order to obtain the $NARMA-L2$ model, the Taylor series expansion of (33) is written down and the first order terms are retained. Hence, the $NARMA-L2$ model can be derived as:

$$\begin{aligned}\hat{y}_{n+d} =& \hat{f}[y_n, y_{n-1}, \cdots, y_{n-k+1}, \\ & u_{n-1}, \cdots, u_{n-k+1}] \\ & + \hat{g}[y_n, y_{n-1}, \cdots, y_{n-k+1}, \\ & u_{n-1}, \ldots, u_{n-k+1}] \cdots u_n\end{aligned} \tag{36}$$

where

$$\begin{aligned}\hat{f} &= F_{actual}[(y_n, \cdots, y_{n-k+1}, 0, \\ & u_{n-1}, \cdots, u_{n-k+1})] \\ \hat{g} &= \left.\frac{\partial F_{actual}}{\partial u_n}\right|_{(y_n, \cdots, y_{n-k+1}; u_n=0, u_{n-1}, \cdots, u_{n-k+1})}\end{aligned} \tag{37}$$

$$\hat{y}_{n+d} = \hat{f}(x_n) + \hat{g}(x_n)u_n \tag{38}$$

where x_n is the input vector, $\hat{f}$ and $\hat{g}$ are nonlinear functions [88]. As can be seen in (38), the functions $\hat{f}$ and $\hat{g}$ are the submodels of the $NARMA-L2$ model and they must be estimated. In the $NARMA-L2$ model, the control input u_n is separated from the nonlinear dynamics and appears linearly. This is the main strength of $NARMA-L2$ model since it provides practicality in control design. Thus $NARMA-L2$ modeling has been used in various researches [88], [95], [96]. Additionally, it provides a good representation for an affine system model. The work proposed in this paper mainly concentrates on designing an inverse optimal controller for non-affine systems. For this purpose, the input-output data is used to obtain the non-affine *NARX* model of the system ($LSSVR_{NARX}$), consequently, this model is utilized to compute the $NARMA-L2$ ($LSSVR_{NARMA-L2}$) submodels. After identifying the dynamics of the system, the estimated submodels ($\hat{f}, \hat{g}$) are used to compute the inverse optimal control law. An illustration of the $NARMA-L2$ model is given in Fig.(2).

After this concise summary of the basic concepts used in this work given above, in the following section the main contribution of this paper, namely the adaptive inverse optimal controller with integrator which has been proposed for the control of nonlinear and non-affine systems is presented in detail.

V. ADAPTIVE INVERSE OPTIMAL CONTROLLER WITH INTEGRATOR

Inverse optimal control has been successfully applied to various nonlinear systems, however, these systems are mainly affine systems where control input appears linearly in system dynamics. The inverse optimal control method has not yet been used for the control of non-affine systems in the literature.

In this paper, the main goal is to explore the applicability of inverse optimal control methodology for non-affine non-linear systems. A novel control method is proposed in order to adopt inverse optimal control technique for the control of a non-affine systems. In the proposed method, a totally adaptive control architecture is constructed where system

identification, optimization and control are executed concurrently at every sampling interval. From the input-output data of the controlled plant, the NARX model of the system is obtained using online LSSVR method. Consequently, this NARX model is converted to a NARMA-L2 model and submodel terms of the dynamics are identified. This step converts a non-affine system to an affine system model. Also, the controller parameters are optimized using the Levenberg-Marquardt algorithm. Finally, the obtained NARMA-L2 model and the optimized controller parameters are utilized together to compute the inverse optimal control law and this is applied to the controlled system. In the subsections that follow, the steps of the control architecture are explained in detail.

A. IDENTIFICATION OF NARX MODEL USING ONLINE LSSVR

The online *LSSVR* method has been used to identify the $NARMA-L2$ model of the system to be controlled. For this purpose, firstly the *NARX* model is obtained, then the $NARMA-L2$ model is attained using this *NARX* model.

The dynamics of a non-linear system can be represented by the nonlinear autoregressive with exogenous inputs (*NARX*) model,

$$y_{n+1} = f([u_n, \cdots, u_{n-n_u}, y_{n-1}, \cdots, y_{n-n_y}]) \tag{39}$$

where u_n is the control input applied to the plant at time n, y_{n+1} is the output of the plant and n_u, n_y represent the number of past control inputs and the number of past plant outputs included in the model respectively [97]. The state vector of the system at time index n is represented as follows:

$$x_n = [u_n, \cdots, u_{n-n_u}, y_{n-1}, \cdots, y_{n-n_y})] \tag{40}$$

Using LSSVR, an estimate of the output of the model can be obtained by employing equations (32), (39) and (40):

$$\hat{y}_{NARX_{n+1}} = \sum_{i=n-L}^{n-1} a_i K(x_n, x_i) + b_n \tag{41}$$

Utilizing inputs $X_n = [x_{n-1}, \cdots, x_{n-L}]$, corresponding outputs of the system $Y_n = [y_n, \cdots, y_{n-L+1}]$, the length of the sliding windows L and langrange multipliers a_i, the training data set (X_n, Y_n) can be obtained. If we assume that $U_n = [\Omega_n + \frac{I}{C}]^{-1}$,where Ω_n is the kernel matrix, C is the penalty term and use equation (9), Lagrange multipliers a_n and bias b_n are obtained as:

$$Y_n = [U_n]^{-1} a_n^T + 1^T b_n \tag{42}$$

$$1a_n^T = 0 \tag{43}$$

Solving the linear system of equations:

$$U_n Y_n = U_n [U_n]^{-1} a_n^T + U_n 1^T b_n \tag{44}$$

$$a_n^T = U_n [Y_n - 1^T b_n] \tag{45}$$

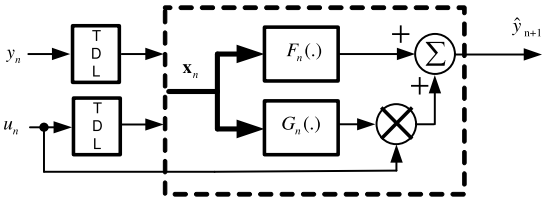


FIGURE 2. $LSSVR_{NARMA-L2}$ model [10], [88]

Using equation (16),

$$1a_n^T = 1U_n[Y_n - 1^T b_n] = 0 \tag{46}$$

$$b_n = \frac{1U_n Y_n}{1U_n 1^T} \tag{47}$$

Thus at time index n :

$$X_n = [x_{n-1}, \cdots, x_{n-L}] \tag{48}$$

$$Y_n = [y_n, \cdots, y_{n-L+1}] \tag{49}$$

At time index $n+1$:

$$X_{n+1} = [x_n, \cdots, x_{n-L+1}] \tag{50}$$

$$Y_{n+1} = [y_{n+1}, \cdots, y_{n-L+2}] \tag{51}$$

At time index $n+1$, new data pair of (x_n, y_n) is added to the training data and old data pair (x_{n-L}, y_{n-L+1}) is discarded from the data set [98], [99].

B. LSSVR BASED NARMA-L2 DECOMPOSITION

After obtaining the *NARX* model of the system, the next step is to decompose it into $NARMA-L2$ submodels. Hence the focus of this step is to obtain the model parameters for $LSSVR_{NARMA-L2}$ from the previously obtained model of the system:

$$\hat{y}_{NARX_{n+1}} = \sum_{i=n-L}^{n-1} a_i K(x_n, x_i) + b_n \tag{52}$$

The *NARX* model which is not directly usable in inverse optimal control design should be separated into submodels $F_n(.)$and $G_n(.)$ as shown in Fig.(2). Submodels can be presented at prediction phase:

$$\hat{f}_n^- \cong F_{n-1}(x_n) \tag{53}$$

$$\hat{g}_n^- \cong G_{n-1}(x_n) \tag{54}$$

Subscript denotes the time index of the state vector and superscript indicates if the system model is derived for time step n or $n-1$. "-" is used to indicate that models computed in the previous time step $n-1$ are evaluated with current state vector x_n with the purpose of obtaining $\hat{f}_n^-$ and $\hat{g}_n^-$. In other words, $\hat{f}_n^+$ and $\hat{g}_n^+$ represent that model acquired in the current

step (n) has been utilized with the current state vector x_n as follows:

$$\hat{f}_n^+ \cong F_n(x_n) \tag{55}$$
$$\hat{g}_n^+ \cong G_n(x_n) \tag{56}$$

To compute inverse optimal control, submodels $F_n(.)$and $G_n(.)$ have to be obtained. The regression functions for $F_{n-1}(x_{n-1})$ and $G_{n-1}(x_{n-1})$ are represented as:

$$\hat{f}_{n-1}^+ \cong F_{n-1}(x_{n-1}) = \sum_{i=1}^{N} \alpha_i k(x_i, x_{n-1}) + b_f \tag{57}$$

$$\hat{g}_{n-1}^+ \cong G_{n-1}(x_{n-1}) = \sum_{i=1}^{N} \theta_i k(x_i, x_{n-1}) + b_g \tag{58}$$

The output of $LSSVR_{NARMA-L2}$ model of the system can be acquired as:

$$\hat{y}_{NARMA_n} = F_{n-1}(x_{n-1}) + G_{n-1}u_{n-1} \tag{59}$$
$$= \sum_{i=1}^{N} \alpha_i k(x_i, x_{n-1}) + b_f + \left[\sum_{i=1}^{N} \theta_i k(x_i, x_{n-1}) + b_g\right] u_{n-1} \tag{60}$$

The models of $LSSVR_{NARMA-L2}$ and $LSSVR_{NARX}$ are matched and the following relation between these two models can be established:

$$\sum_{i=1}^{N} a_i K(x_i, x_{n-1}) + b_n = \sum_{i=1}^{N} [\alpha_i + \theta_i u_{n-1}] K(x_i, x_{n-1}) + b_f + b_g u_{n-1} \tag{61}$$

Hence:

$$a_i = \alpha_i + \theta_i u_{n-1} \tag{62}$$
$$b_n = b_f + b_g u_{n-1} \tag{63}$$

In order to proceed, let us assume that there exists the following relations between submodels:

$$\hat{y}_{NARX_n} \cong \hat{y}_{NARMA_n} \tag{64}$$

The following assumption has been suggested to use this relationship and to estimate the submodels' parameters using this approach:

$$\alpha_i = \mu_1(.)\theta_i \tag{65}$$
$$b_f = \mu_2(.)b_g \tag{66}$$

As a consequence, we obtain:

$$F_{n-1}(x_{n-1}) = \mu_1(.)G_{n-1}(x_{n-1}) + [\mu_2(.) - \mu_1(.)]b_g \tag{67}$$

It is clearly seen from equation (67) that the $LSSVR_{NARMA-L2}$ model of the system depends only on $G_{n-1}(x_{n-1})$ for given values of $\mu_1(.)$ and $\mu_2(.)$, $\mu_1(.)$ and $\mu_2(.)$ are the regressor parameters of the submodels. Using equations (65) and (66), the bias and Lagrange parameters of the $LSSVR_{NARMA-L2}$ submodels $F_{n-1}(x_{n-1})$ and $G_{n-1}(x_{n-1})$ are calculated with respect to $LSSVR_{NARX}$ model:

$$\theta_i = \frac{a_i}{\mu_1(.) + u_{n-1}}, a_i = \mu_1(.)\theta_i$$
$$b_g = \frac{b_n}{\mu_2(.) + u_{n-1}]}, b_f = \mu_2(.)b_g \tag{68}$$

When the transformation from the NARX model (52) to NARMA-L2 (60) is achieved, the biggest advantage obtained is that the control signal term is successfully extricated from the nonlinear terms in the NARX model (52) and transformed into a multiplier term as in NARMA-L2 (60). Thus, an affine nonlinear equivalent of the NARX model can be accomplished. As a result, the control signal term as in (60) can be expressed in terms of the nonlinear dynamics of the NARX model, transformed by Taylor series expansion.

As a consequence, system dynamics that are required for designing an inverse optimal controller can be calculated using the initially obtained $LSSVR_{NARX}$ model and then converting it to a $LSSVR_{NARMA-L2}$ model.

C. PARAMETER OPTIMIZATION USING LEVENBERG-MARQUARDT ALGORITHM

The tracking performance of the proposed method depends on the parameters μ_1, μ_2, P, R and K_I.

To efficiently update all of the aforementioned parameters, a predictive framework that takes into account the system's K-step-ahead future behavior has been deployed.

The objective function E given in equation (69) is optimized by Levenberg-Marquardt algorithm to tune the parameters P, R, μ_1, μ_2 and K_I.

$$E(\mu, P, R, K_I) = \frac{1}{2}\sum_{k=1}^{K} \hat{e}_{n+k}^2 + \frac{1}{2}\lambda[u_n - u_{n-1}]^2 \tag{69}$$

Here, $\hat{e}_{n+k} = r_{n+k} - \hat{y}_{n+k}$, λ is the penalty parameter. Levenberg-Marquardt algorithm is used to adjust the parameters as follows:

$$\begin{bmatrix} \mu 1_{new} \\ \mu 2_{new} \\ P_{new} \\ R_{new} \\ K_{I_{new}} \end{bmatrix} = \begin{bmatrix} \mu 1_{old} \\ \mu 2_{old} \\ P_{old} \\ R_{old} \\ K_{I_{old}} \end{bmatrix} + \left[J^T J + \eta I\right]^{-1} J^T e \tag{70}$$

where (71), as shown at the bottom of the next page.

$e = [\hat{e}_{n+1} \cdots, \hat{e}_{n+K}\sqrt{\lambda}\Delta u_n]^T$ and J is Jacobian matrix. If the jth support vector and current state are $c_n = [u_{n-1}, \cdots, \cdots, u_{n-n_u}, y_n, \cdots, y_{n-n_y+1}]^T$ and $x_j = [x_{j1}, \cdots, x_{jn_u}, x_{jn_u+n_y}]^T$, respectively, K-step ahead future

behavior of the system can be approximated as:

$$\hat{y}_{n+k} = f_n(\hat{c}_{n+k}) + g_n(\hat{c}_{n+k})u_n \tag{72}$$

$$\hat{c}_{n+k} = [\underbrace{u_n, \cdots, u_n}_{\text{k}}, \underbrace{u_{n-1}, \cdots, u_{n+k-n_u}}_{n_u\text{-k}} \underbrace{\hat{y}_{n+k}, \cdots, \hat{y}_{n+1}}_{\text{k-1}} \underbrace{y_n, \cdots, y_{n+k-n_y}}_{n_y+1\text{-k}}] \tag{73}$$

where

$$f_n(\hat{c}_{n+k}) = \sum_{j \in SV} \alpha_j K(d_{n+k,j}) + b_f = \sum_{j \in SV} \alpha_j exp(\frac{-d_{n+k,j}}{2\sigma^2}) \tag{74}$$

$$g_n(\hat{c}_{n+j}) = \sum_{j \in SV} \theta_j K(d_{n+k,j}) + b_g = \sum_{j \in SV} \theta_j exp(\frac{-d_{n+k,j}}{2\sigma^2}) \tag{75}$$

The Euclidean distance between the state vector at step(n+k) and the jth support vector x_j is defined as $d_{n+k,j}$.

The choice of kernel parameter σ significantly influences how well the data can be distinguished in the feature space. With a small σ, the kernel matrix emphasizes similarities among input data, causing similar points in input space to be widely dispersed in feature space. Consequently, this leads to an increase in the number of support vectors needed to accurately capture the system's dynamics. Conversely, for large σ values, dissimilar data in the input space are projected closely in the feature space. This can cause the kernel's nonlinearity to be diminished, resulting in an imprecise identification of the system's dynamics.

$$d_{n+k,j} = \sqrt{(c_{n+k} - x_j)^T (c_{n+k} - x_j)} = \sqrt{D_{U_{n+k}} + D_{Y_{n+k}}} = \sqrt{A_{n+k}} \tag{76}$$

where

$$D_{U_{n+k}} = \sum_{i=1}^{min(k,n_u)} (u_n - x_{j,i})^2 + \sum_{i=min(k,n_u)+1}^{n_u} (u_{n+min(k,n_u)-i} - x_{j,i})^2 - \delta(n_u - k) \tag{77}$$

$$D_{Y_{n+k}} = \sum_{i=1}^{min(k-1,n_y)} (\hat{y}_{n+k-i} - x_{j,n_u+i})^2 + \sum_{i=min(k-1,n_y)+1}^{n_y} (u_{n+min(k-1,n_y)+1-i} - x_{j,n_u+i})^2 - \delta(n_y + 1 - k) \tag{78}$$

$\delta(.)$ represents unit step function. The system Jacobian J_m is given as:

$$\frac{\partial \hat{y}_{n+k}}{\partial u_n} = \frac{\partial f_n(c_{n+k})}{\partial d_{n+k,j}} \frac{\partial d_{n+k,j}}{\partial u_n} + \frac{\partial g_n(c_{n+k})}{\partial d_{n+k,j}} \frac{\partial d_{n+k,j}}{\partial u_n} u_n + g_n(c_{n+k}) \tag{79}$$

where

$$\frac{\partial f_n(c_{n+k})}{\partial d_{n+k,j}} = \frac{-1}{2\sigma^2} \sum_{j \in SV} \alpha_j exp(\frac{-d_{n+k,j}}{2\sigma^2}),$$

$$\frac{\partial g_n(c_{n+k})}{\partial d_{n+k,j}} = \frac{-1}{2\sigma^2} \sum_{j \in SV} \theta_j exp(\frac{-d_{n+k,j}}{2\sigma^2}), \tag{80}$$

$$\frac{\partial d_{n+k,j}}{\partial u_n} = \frac{\partial d_{n+k,j}}{\partial A_{n+k}} \left[\frac{\partial A_{n+k}}{\partial D_{U_{n+k}}} \frac{\partial D_{U_{n+k}}}{u_n} + \frac{\partial A_{n+k}}{\partial D_{Y_{n+k}}} \frac{\partial D_{Y_{n+k}}}{u_n} \right] \tag{81}$$

$$\frac{\partial d_{n+k,j}}{\partial A_{n+k}} = \frac{1}{2\sqrt{A_{n+k}}} = \frac{1}{2d_{n+k,j}}, \tag{82}$$

$$J = \begin{bmatrix} \frac{\partial e_{n+1}}{\partial \hat{y}_{n+1}} \frac{\partial \hat{y}_{n+1}}{\partial \mu_1} & \frac{\partial e_{n+1}}{\partial \hat{y}_{n+1}} \frac{\partial \hat{y}_{n+1}}{\partial \mu_2} & \frac{\partial e_{n+1}}{\partial \hat{y}_{n+1}} \frac{\partial \hat{y}_{n+1}}{\partial P} & \frac{\partial e_{n+1}}{\partial \hat{y}_{n+1}} \frac{\partial \hat{y}_{n+1}}{\partial R} & \frac{\partial e_{n+1}}{\partial \hat{y}_{n+1}} \frac{\partial \hat{y}_{n+1}}{\partial K_I} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_{n+K}}{\partial \hat{y}_{n+K}} \frac{\partial \hat{y}_{n+K}}{\partial \mu_1} & \frac{\partial e_{n+K}}{\partial \hat{y}_{n+K}} \frac{\partial \hat{y}_{n+K}}{\partial \mu_2} & \frac{\partial e_{n+K}}{\partial \hat{y}_{n+K}} \frac{\partial \hat{y}_{n+K}}{\partial P} & \frac{\partial e_{n+K}}{\partial \hat{y}_{n+K}} \frac{\partial \hat{y}_{n+K}}{\partial R} & \frac{\partial e_{n+K}}{\partial \hat{y}_{n+K}} \frac{\partial \hat{y}_{n+K}}{\partial K_I} \\ \frac{\partial \sqrt{\lambda} \Delta u_n}{\mu_1} & \frac{\partial \sqrt{\lambda} \Delta u_n}{\mu_2} & \frac{\partial \sqrt{\lambda} \Delta u_n}{P} & \frac{\partial \sqrt{\lambda} \Delta u_n}{R} & \frac{\partial \sqrt{\lambda} \Delta u_n}{K_I} \end{bmatrix}$$

$$= -\begin{bmatrix} \frac{\partial \hat{y}_{n+1}}{\partial u_n} \frac{\partial u_n}{\partial \mu_1} & \frac{\partial \hat{y}_{n+1}}{\partial u_n} \frac{\partial u_n}{\partial \mu_2} & \frac{\partial \hat{y}_{n+1}}{\partial u_n} \frac{\partial u_n}{\partial P} & \frac{\partial \hat{y}_{n+1}}{\partial u_n} \frac{\partial u_n}{\partial R} & \frac{\partial \hat{y}_{n+1}}{\partial u_n} \frac{\partial u_n}{\partial K_I} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \hat{y}_{n+K}}{\partial u_n} \frac{\partial u_n}{\partial \mu_1} & \frac{\partial \hat{y}_{n+K}}{\partial u_n} \frac{\partial u_n}{\partial \mu_2} & \frac{\partial \hat{y}_{n+K}}{\partial u_n} \frac{\partial u_n}{\partial P} & \frac{\partial \hat{y}_{n+K}}{\partial u_n} \frac{\partial u_n}{\partial R} & \frac{\partial \hat{y}_{n+K}}{\partial u_n} \frac{\partial u_n}{\partial K_I} \\ -\sqrt{\lambda} \frac{\partial u_n}{\mu_1} & -\sqrt{\lambda} \frac{\partial u_n}{\mu_2} & -\sqrt{\lambda} \frac{\partial u_n}{P} & -\sqrt{\lambda} \frac{\partial u_n}{R} & -\sqrt{\lambda} \frac{\partial u_n}{K_I} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{\partial \hat{y}_{n+1}}{\partial u_n} \\ \vdots \\ -\frac{\partial \hat{y}_{n+K}}{\partial u_n} \\ \sqrt{\lambda} \end{bmatrix} \begin{bmatrix} \frac{\partial u_n}{\partial \mu_1} & \frac{\partial u_n}{\partial \mu_2} & \frac{\partial u_n}{\partial P} & \frac{\partial u_n}{\partial R} & \frac{\partial u_n}{\partial K_I} \end{bmatrix} = J_m J_c \tag{71}$$

$$\frac{\partial A_{n+k}}{\partial D_{U_{n+k}}} = \frac{\partial A_{n+k}}{\partial D_{Y_{n+k}}} = 1, \tag{83}$$

$$\frac{\partial D_{U_{n+k}}}{\partial u_n} = \sum_{i=1}^{min(k,n_u)} 2(u_n - x_j, i), \tag{84}$$

$$\frac{\partial D_{Y_{n+k}}}{\partial u_n} = \sum_{i=1}^{min(k-1,n_y)} 2(\hat{y}_{n+k-i} - x_{j,n_u+i}) \frac{\partial \hat{y}_{n+k-i}}{\partial u_n} \delta(k-i) \tag{85}$$

The Jacobian matrix J_m is obtained using the substitutions A and D given in equations (76)-(78) and $LSSVR_{NARMA-L2}$ model of the system as shown in equations (79)-(85). Tracking performance of the system achieved using the proposed method depends on parameters μ_1, μ_2, P, R. Due to the observed steady-state error we also added an integrator to the controller design. Since adaptive parameters are not at their optimal values in the transient state of the closed-loop system, a correction term denoted as δu_n has been added to tolerate the error in the resulting control signal. Using Taylor series expansion given in [100], [101], the correction term can be derived as follows via J_m.

$$\delta u_n = -\left[\mathbf{J}_m^T \mathbf{J}_m\right]^{-1} \mathbf{J}_m^T \hat{\mathbf{e}} \tag{86}$$

The objective function given in equation (69) is optimized by the Levenberg-Marquardt algorithm to tune the parameters. The adjustable controller parameters have also been adapted using the Levenberg-Marquardt algorithm.

D. PROPOSED CONTROL PROCEDURE

A step-by-step outline of the proposed control method is provided below. A flow chart of the pseudo-code that summarizes the algorithm is given in Fig.(3). Additionally, the control architecture of the overall system is illustrated in Fig.(4).

1. Initialize controller and model parameters.
   -Controller parameters: μ_j, P, R, K_I
   -$LSSVR_{NARX}$ model parameters: $\beta = b_\beta = 0$, C = 1000, $\varepsilon = 10^{-3}$, $\sigma = 1$
2. Train $LSSVR_{NARX}$ model using (u_{fn-1}, y_n) data pair
   -Set index to n
   -Constitute $x_{n-1} = [u_{n-2}, \cdots, u_{n-n_u} y_{n-1}, \cdots, y_{n-n_y}]$
   -Predict $\hat{y}_n$ via $LSSVR_{NARX}$ model
   -Calculate $e_{m_n} = y_n$-$\hat{y}_n$
   If $\mid e_{m_n} \mid > \varepsilon$
   Train system model where $e_{m_n} = y_n - \hat{y}_n$ else continue with the system model and controller parameters obtained at the previous step
3. Convert $LSSVR_{NARX}$ model to $LSSVR_{NARMA-L2}$ model
   -Compute parameters of $F_{n-1}(\alpha, b_f)$ and $G_{n-1}(\theta, b_g)$ by using $LSSVR_{NARX}$ parameters (β, b_β) as shown in (68)
   -Constitute $x_n = [u_{n-1}, \cdots, u_{n-n_u}, y_n, \cdots, y_{n-n_y+1}]$
   -Compute $\hat{f}_n^-$ and $\hat{g}_n^-$ by $F_{n-1}(.)$ and $G_{n-1}(.)$
4. Calculate the control input produced by Inverse Optimal Controller with integrator (u_n).
   -Calculate the control signal (u_n) using the predictions of submodels and control law in (20)
5. Apply control signal u_n to the system to compute y_{n+1} -Training data pair of $LSSVR_{NARX}$ model (u_n, y_{n+1}) for next step is obtained.
6. Calculate the Jacobian
   -Apply u_n K-times to $LSSVR_{NARX}$ model in order to constitute Jacobian matrix in (71)
7. Learning Step for Controller
   -Update Controller parameters (μ_j, P, R, K_I) using (70)
8. Increment n = n + 1

VI. SIMULATION RESULTS

The proposed control methodology has been assessed with simulations performed on two different nonlinear benchmark problems. The control system performance has been evaluated for the nominal case and for the cases with measurement noise, input disturbance and parametric uncertainty.

Assuming that the system dynamics are unknown, we initially used online $LSSVR_{NARX}$ to detect the unknown dynamics using input-output data pairs.

Then by the proposed method, $LSSVR_{NARMA-L2}$ model is obtained as given in equations (61-68, 70).

For both benchmark problems, we used staircase type of reference signals for the systems to track. The reason that we chose such reference signals is that they are rather difficult to follow and also transient responses and steady-state errors can be clearly observed. In order to make the simulations as realistic as possible, we followed a hypothetical approach where we assumed that we do not have any prior information about the dynamics of the system to be controlled, however, the real system model is given by the mathematical equations of the benchmark systems. In other words, we assume that we can use the computations of the mathematical dynamical equations of the benchmark systems as if they are measurements obtained from the physical plants. At every iteration the system output is computed using the system dynamics given with equation (89) for benchmark system I and equation (90) for benchmark system II. The input and output values are used to construct the input feature vector to $LSSVR_{NARX}$ which computes the NARX model of the

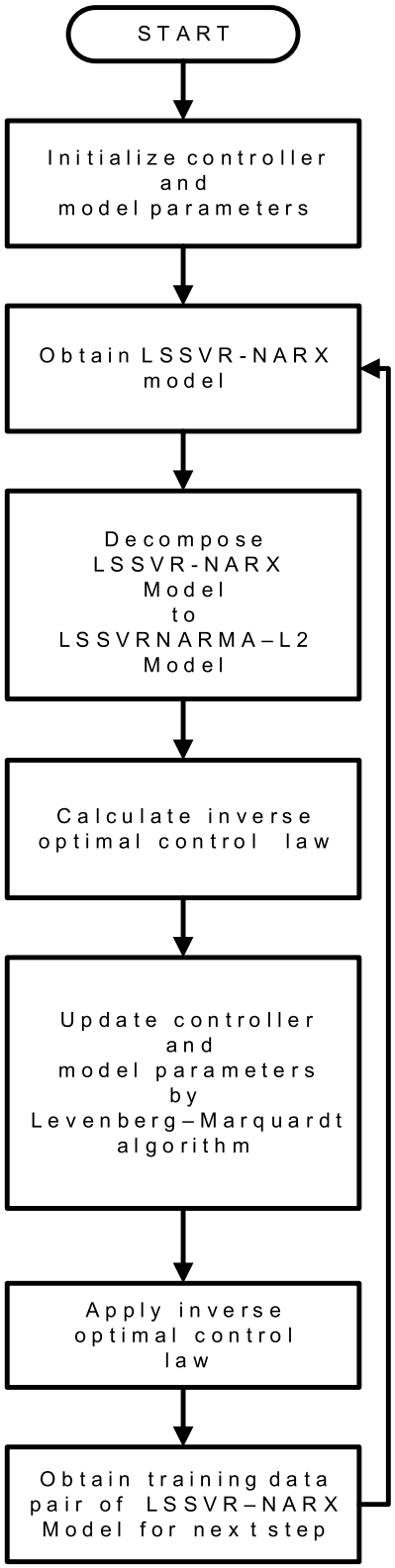


FIGURE 3. Flow Chart of the Proposed Method.

system. Then the NARX model is converted to submodels $\hat{f}$ and $\hat{g}$ by $LSSVR_{NARMA-L2}$. The parameters P, R, μ_1, μ_2 and K_I are optimized using Levenberg-Marquardt algorithm with equation (70). The computed submodels and the optimized parameters are utilized to calculate the inverse optimal control law which was given by equation (20). Consequently, the control law is applied to the actual system, a new system output value is computed for the applied control input. With the new control input and system output the next iteration of the simulation starts to recompute the NARX model, convert it to submodules, optimize controller parameters and calculate the inverse optimal control law. Obviously, at every iteration system identification and control are carried out together and in an adaptive manner.

In order to verify the effectiveness of the proposed control method, the simulations are repeated for two different benchmark systems which are nonlinear and non-affine. Also, to assess the robustness of the method simulations are repeated for the cases where measurement noise, input disturbance or parametric uncertainty are added to the system. The simulation scenario summarized above is applied to observe how the proposed method deals with these cases and whether it achieves robustness. Hence, for both benchmark systems I and II, simulations have been performed for the following cases:

1) The nominal case.
2) Measurement noise is added to the system.
3) Input disturbance is added to the system.
4) Parametric uncertainty is added to the system.

The input feature vectors of $LSSVR_{NARX}$ and $LSSVR_{NARMA-L2}$ models are chosen as $M_c = [u_{n-1}, \cdots, u_{n-n_u}, y_n, ..y_{n-n_y+1}]^T$ where $n_u = 3$ and $n_y = 3$ represent the number of past inputs and outputs. Since the inverse optimal controller parameters are optimized based on a predictive objective function in equation (69), the grid search algorithm is used for obtaining the optimal estimation horizon (K) and penalty term (λ). The application of this algorithm yielded the optimum K and λ values for system 1 and system 2 as $K = 1$, $\lambda = 0.1000$ and $K = 5$, $\lambda = 0.2000$, respectively. The 3-D error surfaces obtained as a result of the grid search algorithm are illustrated in Fig.(5) and Fig.(6) for system 1 and system 2, respectively.

The derivation of the inverse optimal controller involves the utilization of a positive symmetric Lyapunov function. Consequently, the control law is formulated so that the negative definiteness of the derivative of the Lyapunov function is assured. Hence, it is guaranteed that the inverse optimal controller is inherently stable as long as P and R matrices are appropriately selected. In simulation results, we also verified the stability of each case by plotting the Lyapunov function and its derivative.

A. BENCHMARK SYSTEM I

1) NOMINAL CASE

Firstly, the performance of the proposed method has been tested on the following non-affine nonlinear system described by:

$$x_{1_{n+1}} = 0.1x_{1_n} + \frac{2(u_n + x_{2_n})}{(1 + (u_{n+1} + x_{2_n}^2)} \tag{87}$$

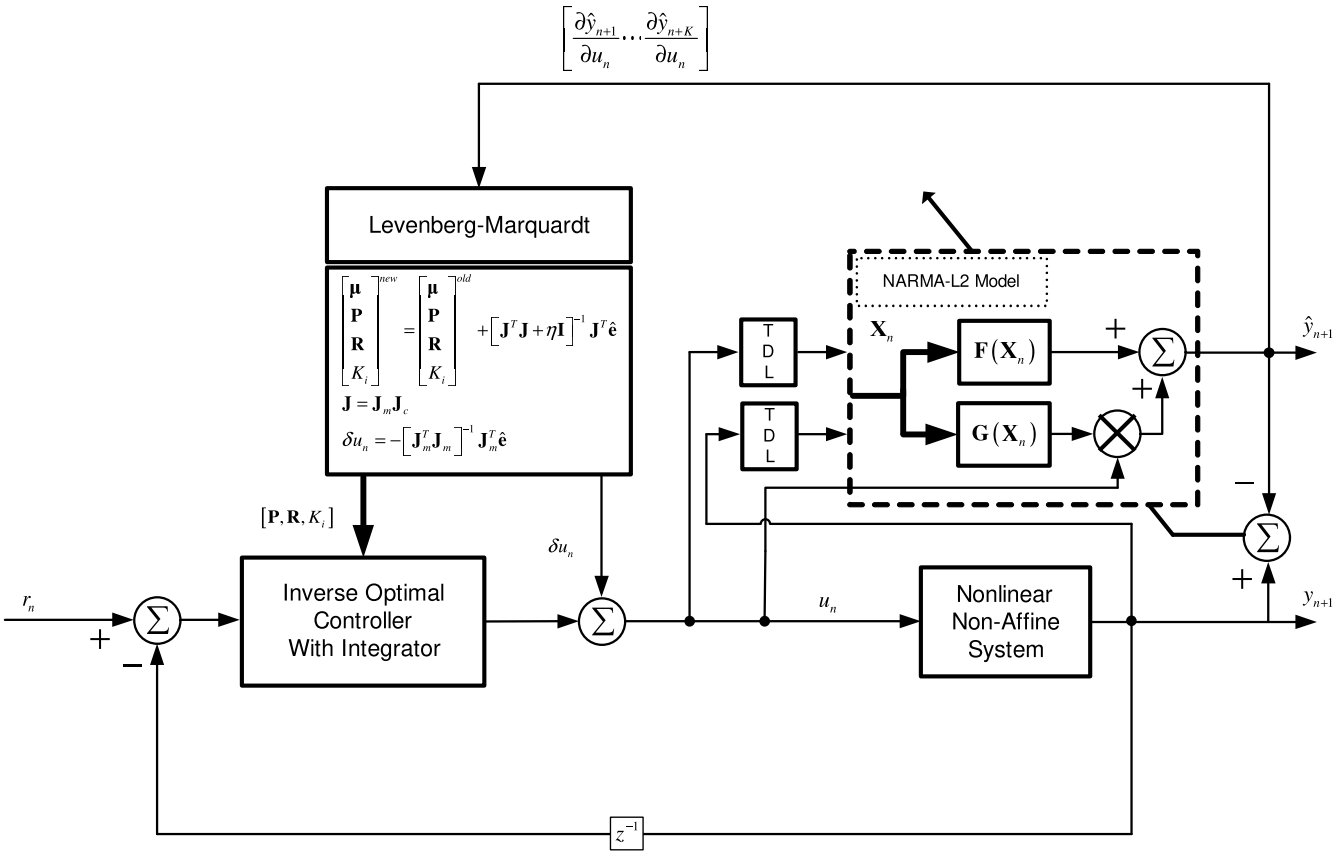


FIGURE 4. Adaptive LSSVR based Inverse Optimal Controller Scheme.

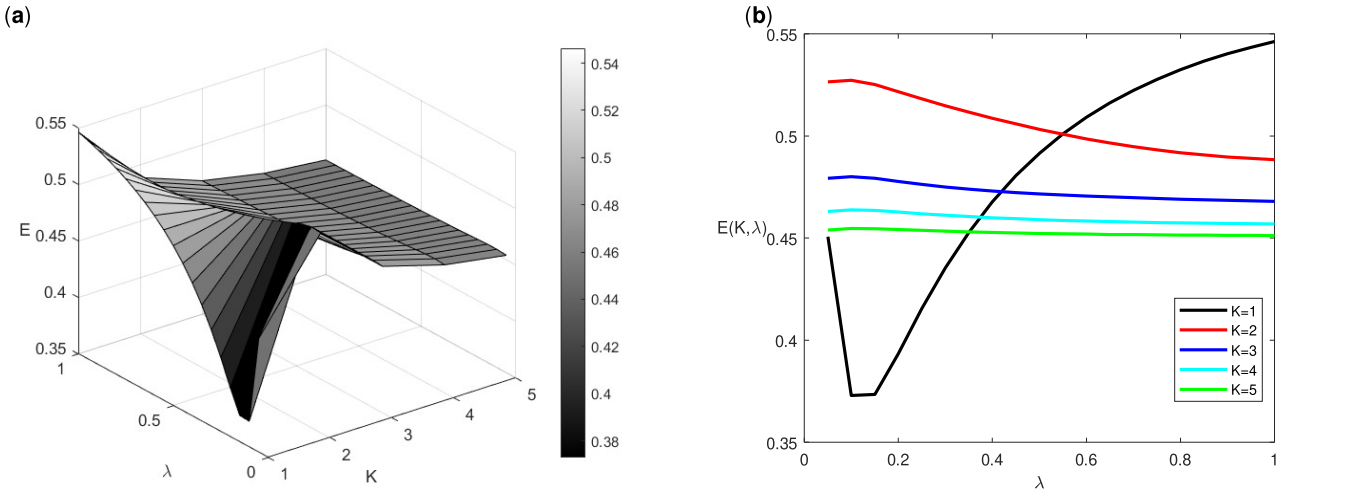


FIGURE 5. 3-D Error surface for Benchmark System I via Grid Search.

$$x_{2_{n+1}} = 0.1x_{2_n} + u_n \frac{(2+u_n^2)}{(1+x_{1_n}^2+x_{2_n}^2)} \tag{88}$$

$$y_{n+1} = x_{1_{n+1}} \tag{89}$$

where initial parameters are assigned as $P(0) = 0.000120$, $R(0) = 0.40$, $K_I(0) = 0.01$.

Fig.(7) and Fig.(8) illustrate the tracking performance results for the nominal case. In Fig.(7) the states, control input law and the control signal correction term are depicted. In Fig.(8), P, Q and K_I parameters and μ_1 and μ_2 are illustrated. Fig.(9) depicts the trajectory tracking error, the Lyapunov function and its derivative. From the figures, it can be observed that

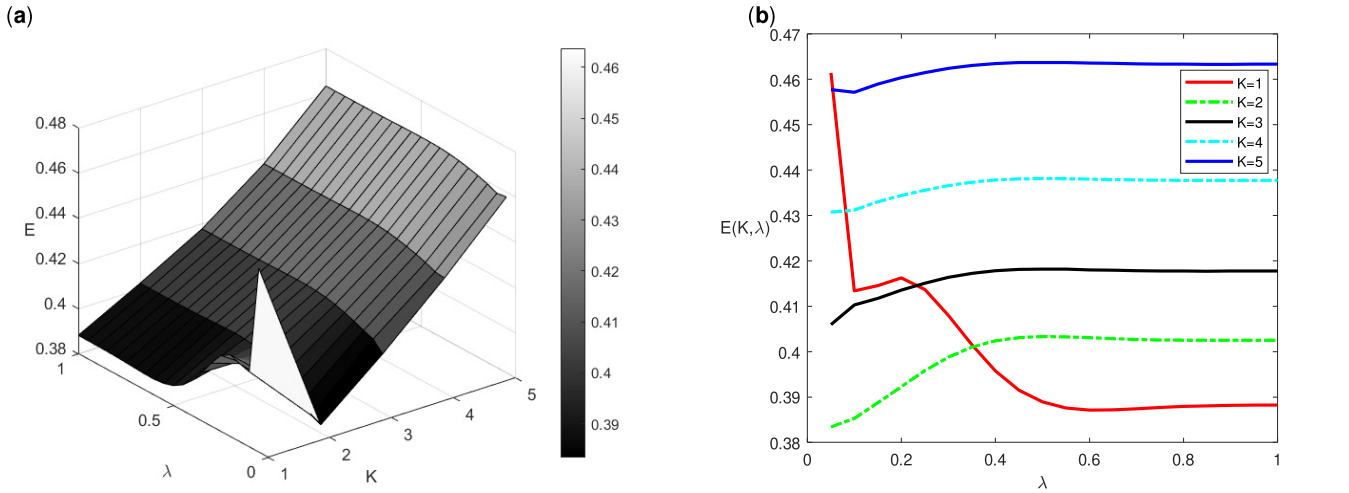


FIGURE 6. 3-D Error surface for Benchmark System II (CSTR) via Grid Search.

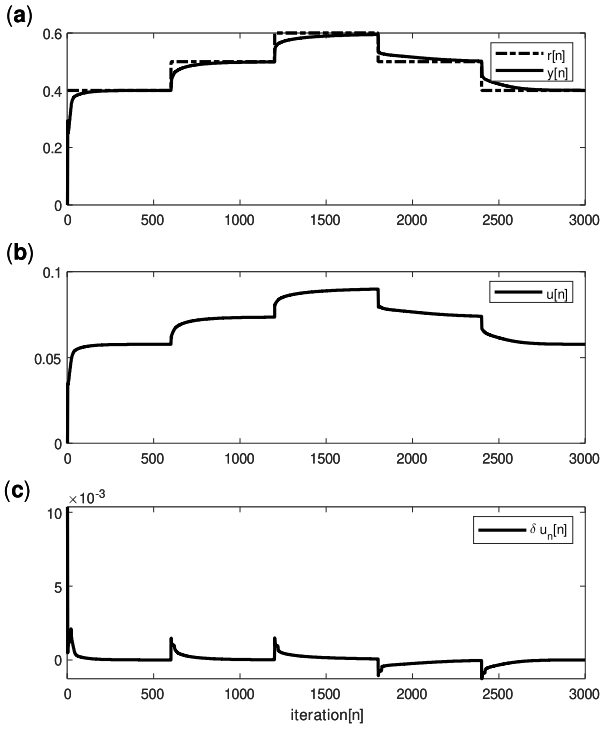


FIGURE 7. Tracking performance (a), Inverse optimal control signal (b), and control signal correction term (c) for Benchmark System I (nominal case).

the trajectory error has a very small magnitude. The output converges to the reference signal fast and tracks it with minimum steady-state error. The parameters tune themselves adaptively to provide successful tracking. Also it can be concluded that the Lyapunov function is always positive and the derivative of the Lyapunov function is always negative, proving the stability of the proposed controller for benchmark system I for the nominal case.

2) MEASUREMENT NOISE CASE

To evaluate the trajectory tracking performance of the proposed control scheme under measurement noise, 30 dB measurement noise is added to the output of the benchmark system I. The system output, calculated control input and control signal correction term can be observed in Fig.(10). Fig(11) depictures the parameters of the inverse optimal controller with integrator and $LSSVR_{NARMA-L2}$ decomposition parameters. It can be clearly seen that the proposed controller can successfully deal with the measurement noise and the parameters adapt themselves to perform successful control.

Fig.(12) depicts the trajectory tracking error, the Lyapunov function and its derivative. It is observed that the proposed controller guarantees stability for benchmark system I under measurement noise. The figures clearly show that despite the presence of measurement noise, output trajectory successfully follows the reference signal and there is nearly no steady-state error. Naturally, because of the measurement noise added to the system oscillations are observed, but these are small magnitude oscillations and the proposed control method can successfully deal with them.

3) INPUT DISTURBANCE CASE

Simulations are repeated for the case where an input disturbance of $u_{d_n} = 0.01\sin(\frac{2\pi}{12.5}n)$ is added. The tracking performance is given in Fig.(13). The evolution of the inverse optimal controller and decomposition parameters are depictured in Fig.(14). The figures clearly show that the proposed control method can successfully provide good control and also can effectively deal with input disturbance. Although there are small magnitude oscillations in the response, the output signal successfully follows the reference signal and the observed steady-state error is very small. It can be clearly seen how the computed control input signal varies in an oscillatory manner to deal with the disturbance added

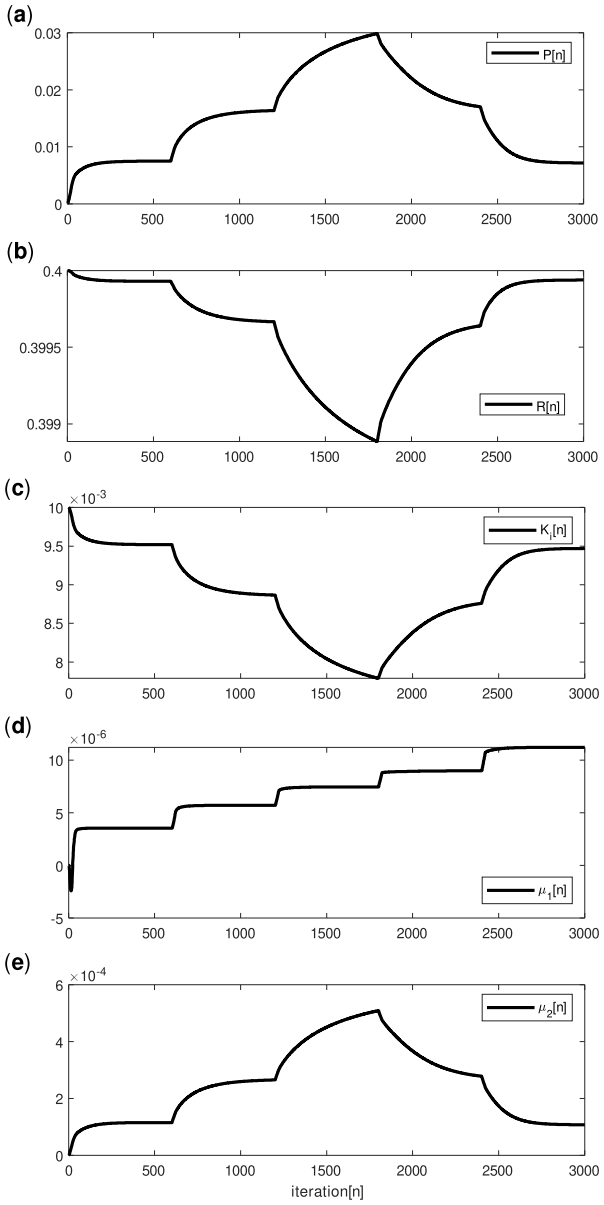


FIGURE 8. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System I (nominal case).

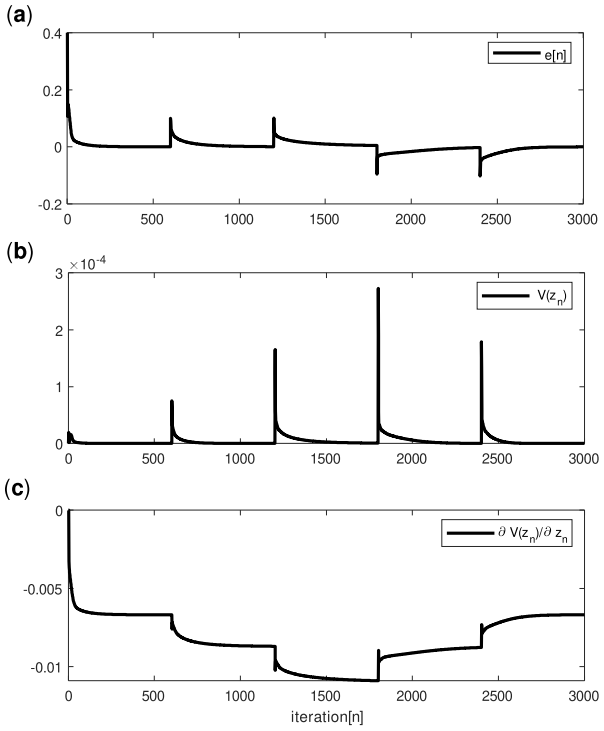


FIGURE 9. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System I (nominal case).

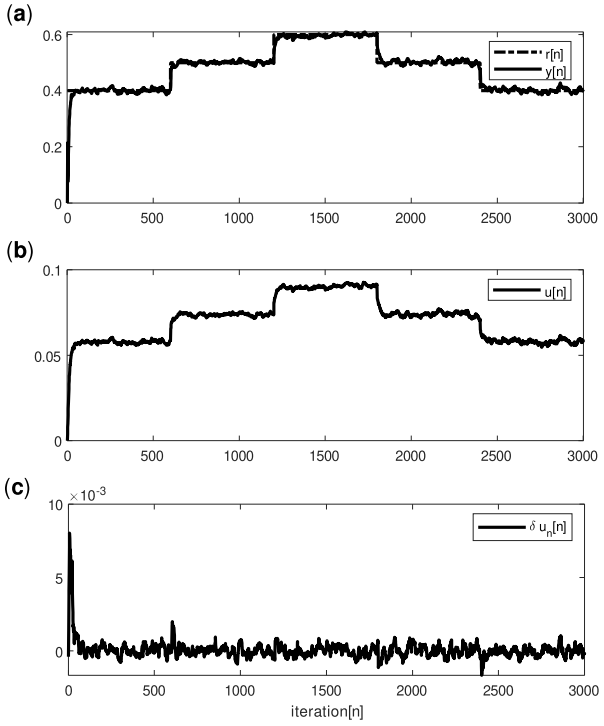


FIGURE 10. Tracking performance (a), Inverse optimal control signal (b), and control signal correction term (c) for Benchmark System I (measurement noise case).

to the system. The adaptive and model-based nature of the proposed control method makes it possible to easily vary the control input signal to minimize the error. Additionally, Fig.(15) visualizes the simulation results for the trajectory tracking error and stability acquired by the proposed method for the benchmark system I under input disturbance.

4) PARAMETRIC UNCERTAINTY CASE

The coefficient of the term "$0.1x_{1_n}$" in (87) is considered as an uncertain parameter given as $p_n x_{1_n}$ where $p_n = 0.1(1 + 0.1sin(2\pi \frac{1}{25}n))$. The tracking control performance and the change of the uncertain parameter are depictured in Fig.(16). Inverse optimal controller with integrator parameters and $LSSVR_{NARMA-L2}$ decomposition parameters are illustrated in Fig(17). The figures display very good tracking results with very small steady-state error. Parametric uncertainty is introduced to the system, however with the proposed

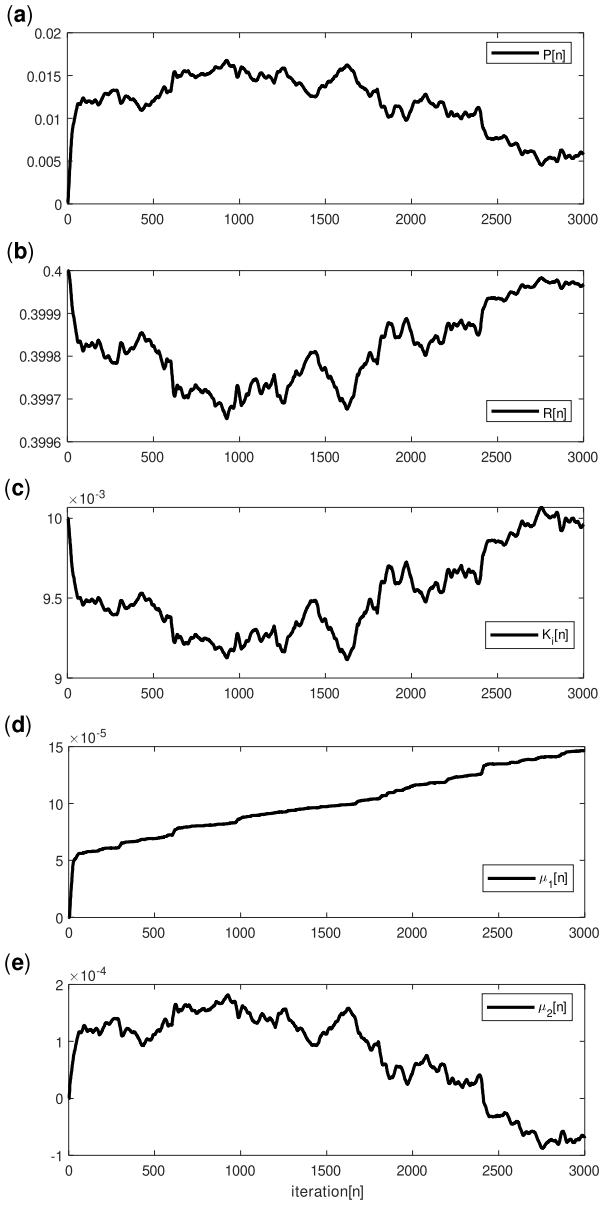


FIGURE 11. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System I (measurement noise case).

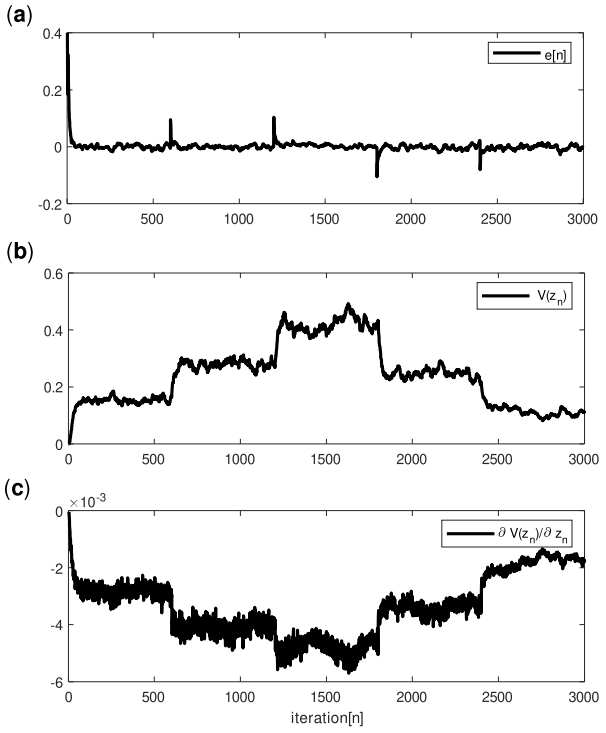


FIGURE 12. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System I (measurement noise case).

control method the output of the benchmark system I can successfully follow the reference signal with slight oscillations. The steady-state error in the output is very small. As can be clearly seen from these figures, the controller can provide successful control despite parametric uncertainty. The controller parameters optimize themselves to achieve effective control.

Fig.(18) demonstrates the trajectory tracking error and stability of benchmark system I when parametric uncertainty is added to the system.

B. BENCHMARK SYSTEM II

1) NOMINAL CASE

The performance of the proposed method has also been tested on a third-order continuously stirred tank reactor (CSTR) system. Continuously stirred tank reactors (CSTRs) are frequently utilized in continuous hydrogen production [102]. While chemicals flow into the reactor to undergo a reaction, there is also an exit stream to extract chemicals from the reactor to keep the volume constant. So, the main advantage of CSTRs is that they provide continuous production without filling and emptying the reactor repeatedly [103]. The dynamical equations of the CSTR system are highly nonlinear and non-affine. Since the control of the CSTR system is a difficult task, it has frequently been employed as a benchmark problem to examine the performances of proposed control methods [101], [104]. The differential equations describing the system are given as [101], [103], [104]:

$$\begin{aligned}\dot{x}_1(t) &= 1 - x_1(t) - Da_1 x_1(t) + Da_2 x_2^2(t)\\ \dot{x}_2(t) &= -x_2(t) + Da_1 x_1(t) - Da_2 x_2^2(t)\\ &\quad - Da_3 d_2(t) x_2^2(t) + u(t)\\ \dot{x}_3(t) &= -x_3(t) + Da_3 d_2(t) x_2^2(t)\end{aligned} \tag{90}$$

where $Da_1 = 3$, $Da_2 = 0.5$, $Da_3 = 1$, $d_{2_{\text{nominal}}}(t) = 1$. The controlled input of the system is $u(t)$ and $x_3(t)$ is the output of the process. This model of the CSTR system is a nonlinear affine system, in order to test the superiority of our proposed method for nonlinear non-affine systems, the system input signal term $(u(t))$ in equation (90) has been modified as $sin(\frac{2\pi}{4}u^2(t))$. Thus, a non-affine system has been obtained to examine the performance of the introduced control architecture. We have used 4th order Runge-Kutta

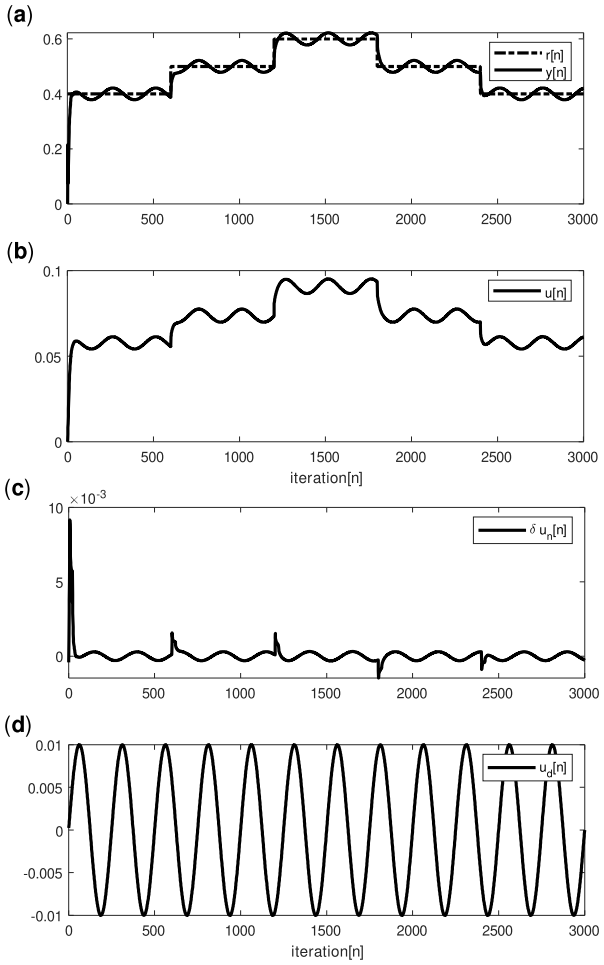


FIGURE 13. Tracking performance(a), Inverse optimal control signal (b), and control signal correction term (c), input disturbance (d) for Benchmark System I (input disturbance case).

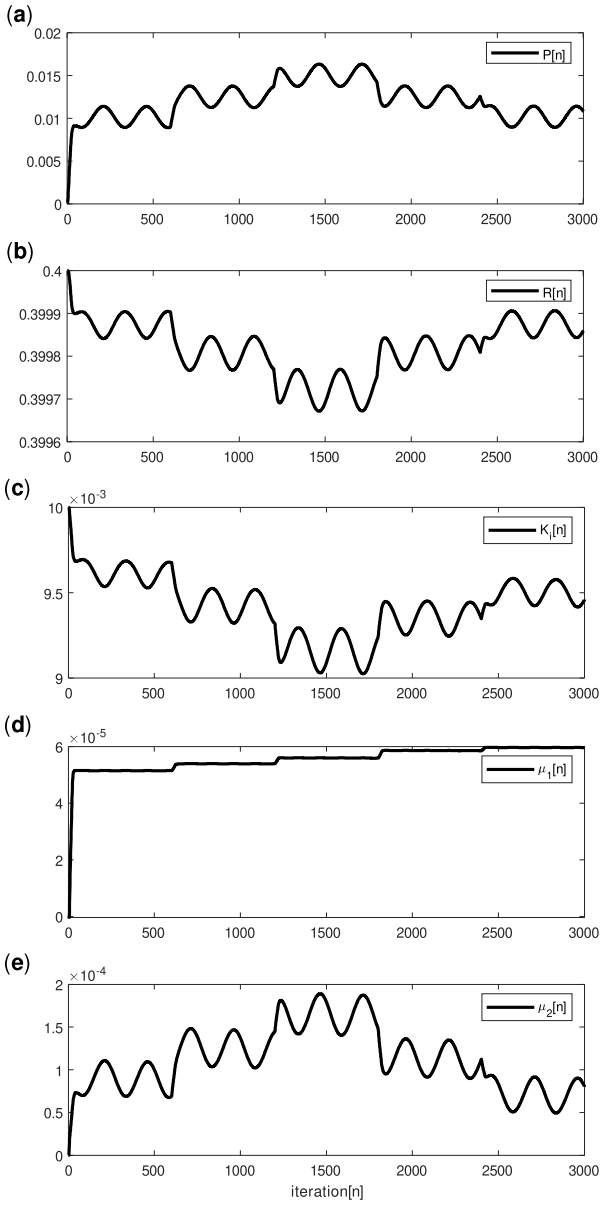


FIGURE 14. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System I (input disturbance case).

method with sampling time $T_s = 0.05$ sec. to discretize the system model [100]. It must be noted that continuous systems can be discretized in accordance with the Nyquist sampling theorem and consequently discrete-time controllers can be utilized.

Initial parameters for the controller are set as $P(0) = 0.58$, $R(0) = 0.04$, $K_I(0) = 0.01$. In the simulation results given below, the results for the nominal case, and for the cases where measurement noise, input disturbance and parametric uncertainty added to the system are depicted. For each case, the tracking results for the output, the computed control input, control signal correction term, the inverse controller and integrator parameters, $LSSVR_{NARMA-L2}$ decomposition parameters and trajectory tracking error are illustrated.

Additionally, the Lyapunov function and its derivative are also provided to prove the stability of the proposed method.

In Fig.(19) and Fig(20) the trajectory tracking results for the nominal case with no noise, disturbance or parametric uncertainty are given. As can be observed from the figures, the proposed method provides good tracking performance with fast response and very small steady-state error. The output of the system displays very short transients and quickly converges to the reference signal. It follows the reference signal with a negligible amount of steady-state error after converging. The figures also show how the controller parameters adjust themselves adaptively to achieve good tracking performance with minimum error. Fig.(21) illustrates that trajectory tracking error is minimized, also the Lyapunov function is always positive and its derivative is always negative, which justifies that the proposed method guarantees the stability of the CSTR system.

2) MEASUREMENT NOISE CASE

The performance of the proposed method has been tested when 30 dB measurement noise is added to the output

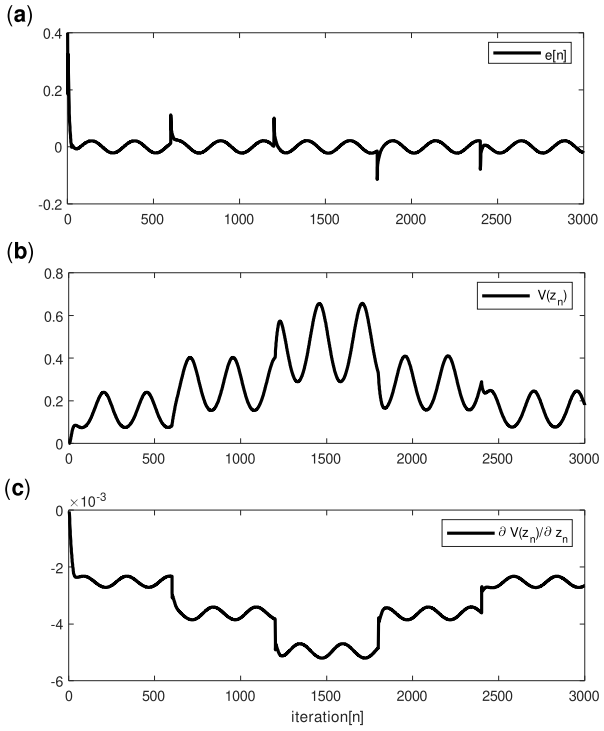


FIGURE 15. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System I (input disturbance case).

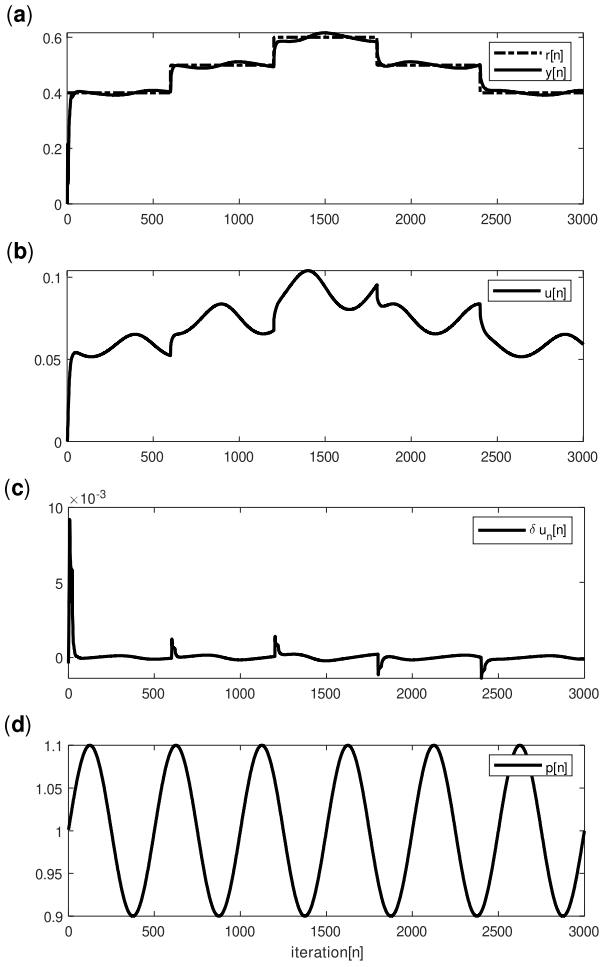


FIGURE 16. Tracking performance(a), Inverse optimal control signal(b), control signal correction term (c), and uncertain system parameter (d) for Benchmark System I (parametric uncertainty case).

of the benchmark system II (CSTR). Fig.(22) represents the tracking performance of the proposed method, Fig.(23) shows the parameters of inverse optimal controller with integrator and parameters obtained for the $LSSVR_{NARMA-L2}$ system identification for the case with measurement noise. The measurement noise added to the system naturally results in oscillations in the system output. However, despite oscillations, it can be observed that the output of the system successfully follows the reference signal. The proposed adaptive control method results in a similar oscillatory control input. To suppress noise, the control input signal adapts itself continuously yielding a good tracking performance with small transient effects, fast convergence and very small steady-state error. The simulation results justify that the proposed method yields satisfactory results when there is measurement noise in the system.

Fig.(24) depicts the signs of the Lyapunov function and its derivative assuring stability and justifies the minimization of the trajectory tracking error for the CSTR system under measurement noise.

3) INPUT DISTURBANCE CASE

In order to evaluate the performance of the proposed method under input disturbance for the CSTR system, an input disturbance of $d(t) = 0.01\sin(\frac{2\pi}{12.5}t)$ is implemented. Fig.(25) clearly shows the success of the tracking performance for the proposed method under disturbance. The system output successfully tracks the reference signal with short transients and nearly no steady-state error. The sinusoidal input disturbance applied to the system inevitably causes oscillations with small magnitudes. However, the adaptive nature of the proposed control method can effectively deal with the disturbance as observed with the oscillatory and changing behavior of the control input signal which is computed. Fig.(26) illustrates the adaptation of inverse optimal controller with integrator and $LSSVR_{NARMA-L2}$ system model decomposition parameters. The stability and the tracking error results are given in Fig.(27) for the case when input disturbance is added to the CSTR system.

4) PARAMETRIC UNCERTAINTY CASE

The performance of the proposed control algorithm has also been assessed under the presence of parametric uncertainty in the system. For this purpose, it has been assumed that there is a sinusoidal uncertainty in parameter $d_2(t)$ which varies as $d_2(t) = (1 + 0.1\sin 0.08\pi t)$. The results obtained for the CSTR system verify that the proposed method is very successful at suppressing parametric uncertainties. Tracking performance is illustrated in Fig.(28). The adaptation of the parameters of the inverse optimal controller with integrator

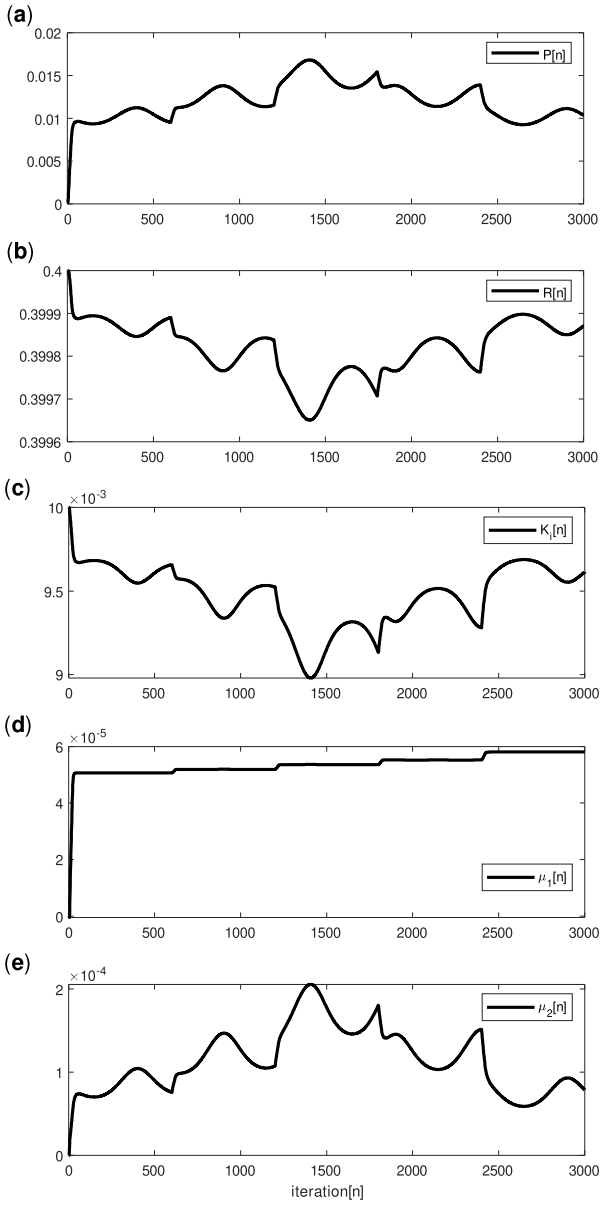


FIGURE 17. Adaptive P (a), R (b), and K_I (c) parameters, Adaptive μ_1 (d), and μ_2 (e) parameters for Benchmark System I (parametric uncertainty case).

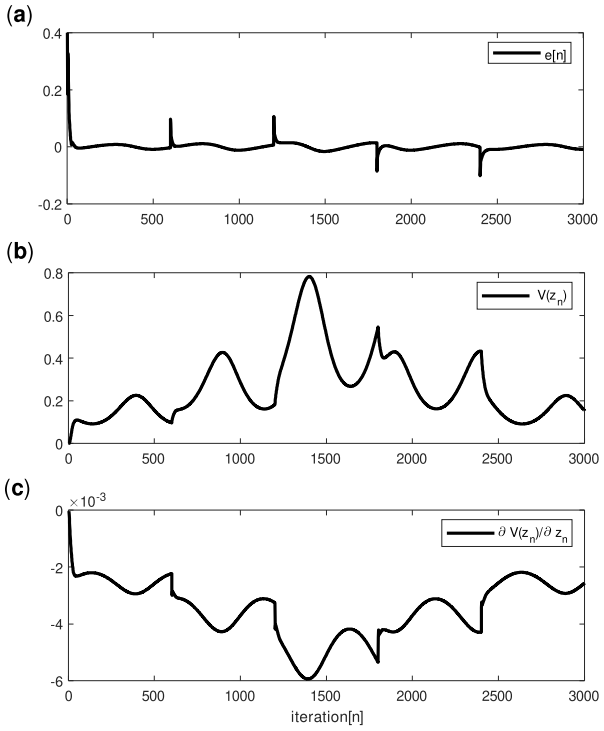


FIGURE 18. Trajectory Tracking Error(a), Lyapunov Function (b), and its derivative (c) for Benchmark System I (parametric uncertainty case).

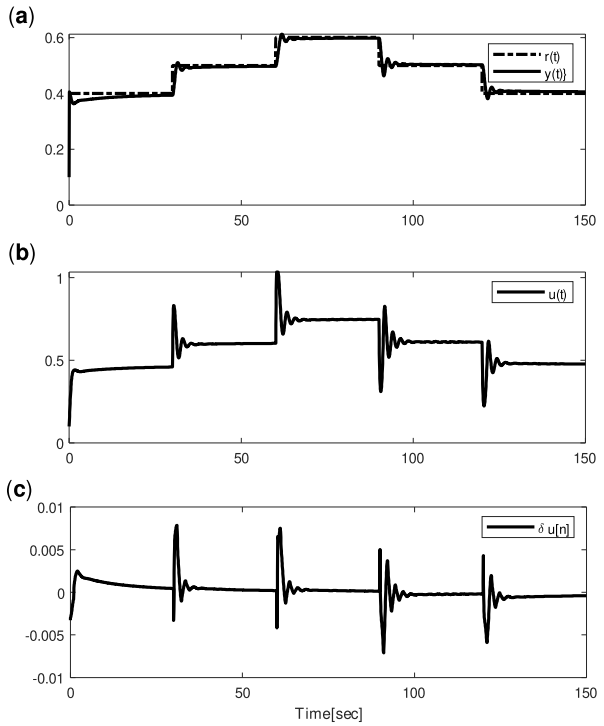


FIGURE 19. Tracking performance (a), Inverse optimal control signal (b), and control signal correction term (c) for Benchmark System II (CSTR) (nominal case).

and $LSSVR_{NARMA-L2}$ system identification parameters can be observed at Fig.(29). The figures display that the system output converges to the reference signal fast and tracks it very successfully. The transient effects vanish quickly and the steady-state error is very small. The adaptation of the controller parameters and the control input signal yields good control performance despite parametric uncertainty in the system. Fig.(30) provides the trajectory tracking error. It also depicts the Lyapunov function and its derivative to prove the stability of the proposed control approach for the case when parametric uncertainty is introduced to the CSTR system.

The simulation results justify that the proposed control method is successful in providing good trajectory control with fast vanishing and small magnitude transient effects and small steady-state errors. The robustness of the method is also verified since the method assures good performance

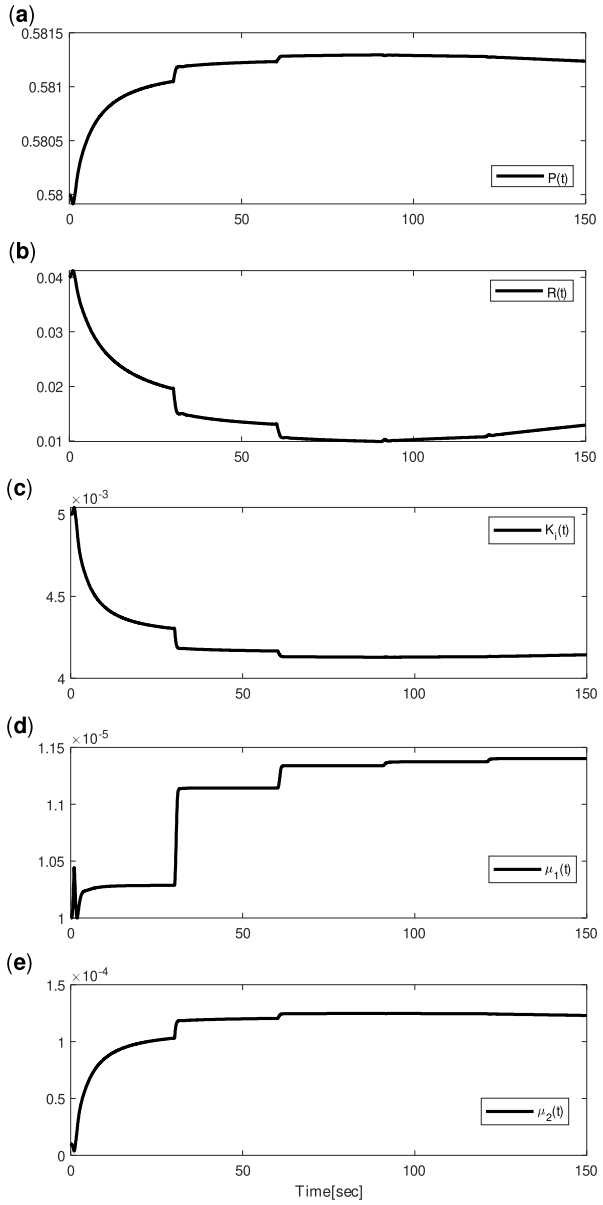


FIGURE 20. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System II (CSTR) (nominal case).

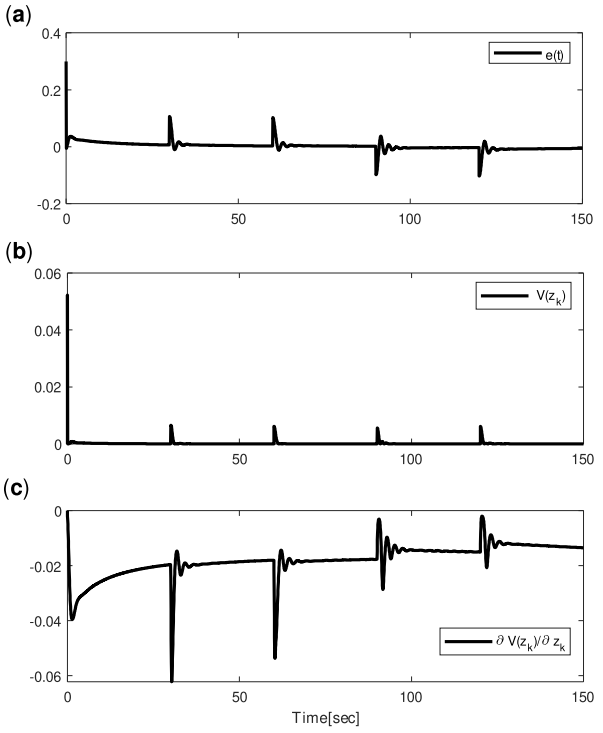


FIGURE 21. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System II (CSTR) (nominal case).

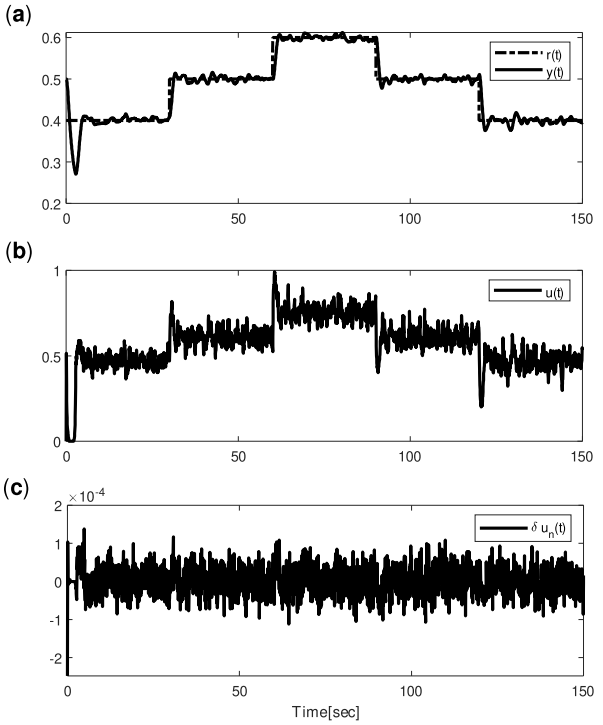


FIGURE 22. Tracking performance (a), Inverse optimal control signal (b), and control signal correction term (c) for Benchmark System II (CSTR) (measurement noise case).

when measurement noise, input disturbance or parametric uncertainty are imposed on the system. The main reason for the good performance attained by the proposed control method is that it is a model-based and totally adaptive approach. At each sampling time, the dynamics of the system to be controlled are identified. The controller parameters are optimized using the Levenberg-Marquardt algorithm. The information of the obtained system model and the optimized controller parameters are utilized in computing the inverse optimal control law. Identification and control are carried out concurrently at each time sample. Hence, the control law can instantly take an action for any change in the reference trajectory, system dynamics and for the noise, disturbance or uncertainty effect that are observed. Consequently, the effects of these factors on the control performance are diminished before they build up. Therefore, we can conclude that the adaptive nature and the optimization property of the proposed control method assure good control performance.

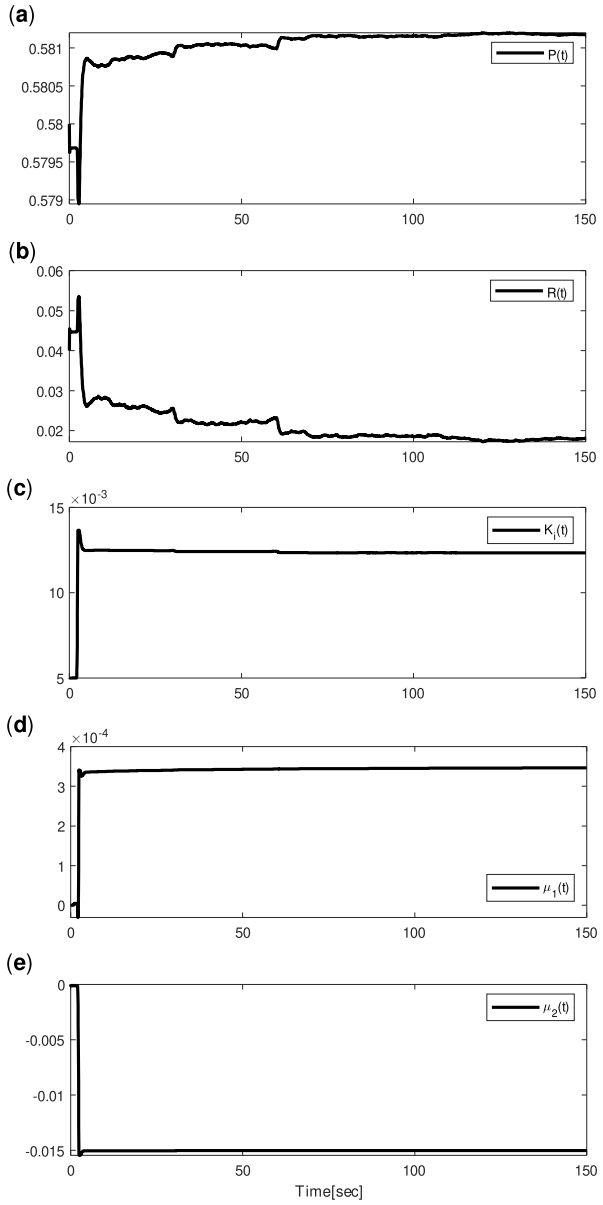


FIGURE 23. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System II (CSTR) (measurement noise case).

VII. COMPARISON OF THE PROPOSED CONTROL APPROACH WITH PID CONTROLLER

We have compared the performance of the adaptive LSSVR based inverse optimal controller with integrator for nonlinear non-affine systems proposed in this paper with a conventional PID controller. As in the previous section staircase type of reference input signal has been used to observe the transient effects and steady-state error clearly. A detailed comparison has been carried out for both of the benchmark systems and robustness analysis has also been performed. For a meaningful comparison, the integral gain that has been computed by the proposed method has been utilized. The proportional and derivative gains of the PID controller have been determined in accordance with the Zieglers-Nichols method, utilizing the value of the integral gain [105].

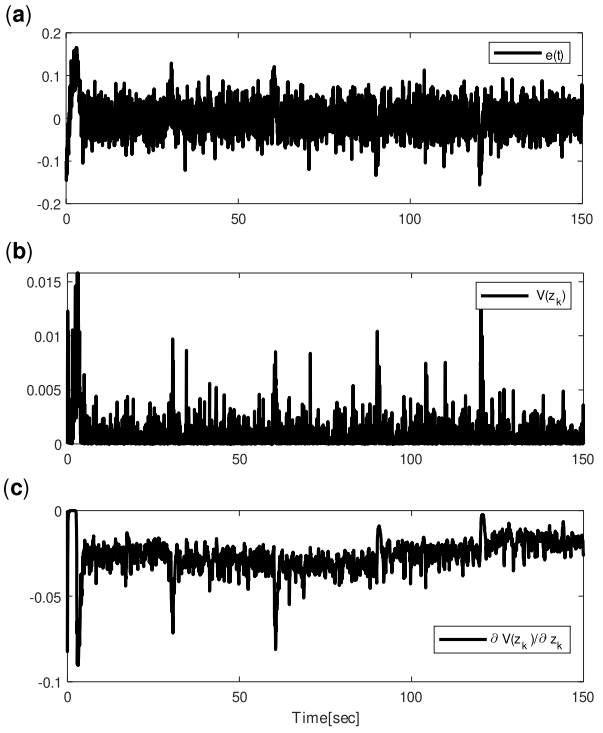


FIGURE 24. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System II (CSTR) (measurement noise case).

We have carried out a detailed comparison for both of the benchmark systems, for the nominal case and for the cases when measurement noise, input disturbance and parametric uncertainty are added to the systems.

$$\text{ISE} = \int \text{e}^2(\text{t})\text{dt} \tag{91}$$

To verify the effectiveness of the proposed approach with respect to PID, the ISE index is computed for each simulated case and for both benchmark problems, all of the results are tabulated.

A. BENCHMARK SYSTEM I

For the benchmark system I given by (89), performance results are tabulated in Table 1. In the table, the ISE performance index is given for all the simulated cases, namely for the nominal case and for the cases with measurement noise, input disturbance and parametric uncertainty. It can be observed from the table that the proposed control method yields better tracking performance than PID controller for all the cases. The gain parameters of the PID controller for all the cases are chosen as $K_p = 0.01$, $K_i = 0.0095$, $K_d = 0.026$. The integral gain is set as the same value that was utilized in the integral part of the proposed control method. The proportional and derivative gains are determined in accordance with the integral gain using Ziegler-Nichols table [105]. It is clearly seen that results obtained by the proposed method are far better in cases with noise, input disturbance and parametric uncertainty.

TABLE 1. Comparison of proposed control method and PID controller with respect to ISE performance index(System in (82)).

Controller Type \ Cases	Nominal	Measurement Noise	Disturbance	Uncertainty
Inverse	0.0489	0.0476	0.0729	0.0494
PID	0.1194	0.2652	0.9898	0.9183

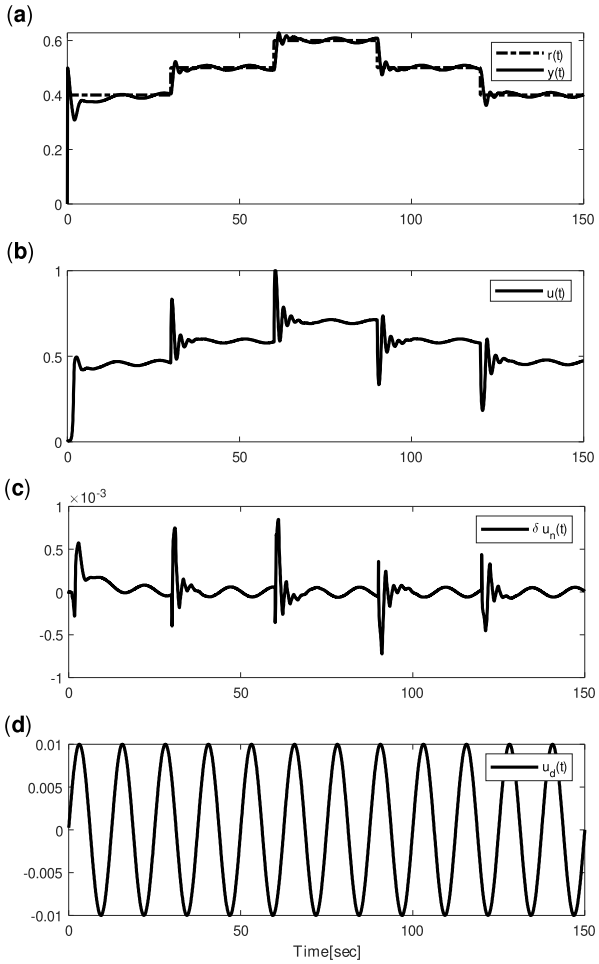


FIGURE 25. Tracking performance (a), Inverse optimal control signal (b), and control signal correction term (c), input disturbance (d) for Benchmark System II (CSTR) (input disturbance case).

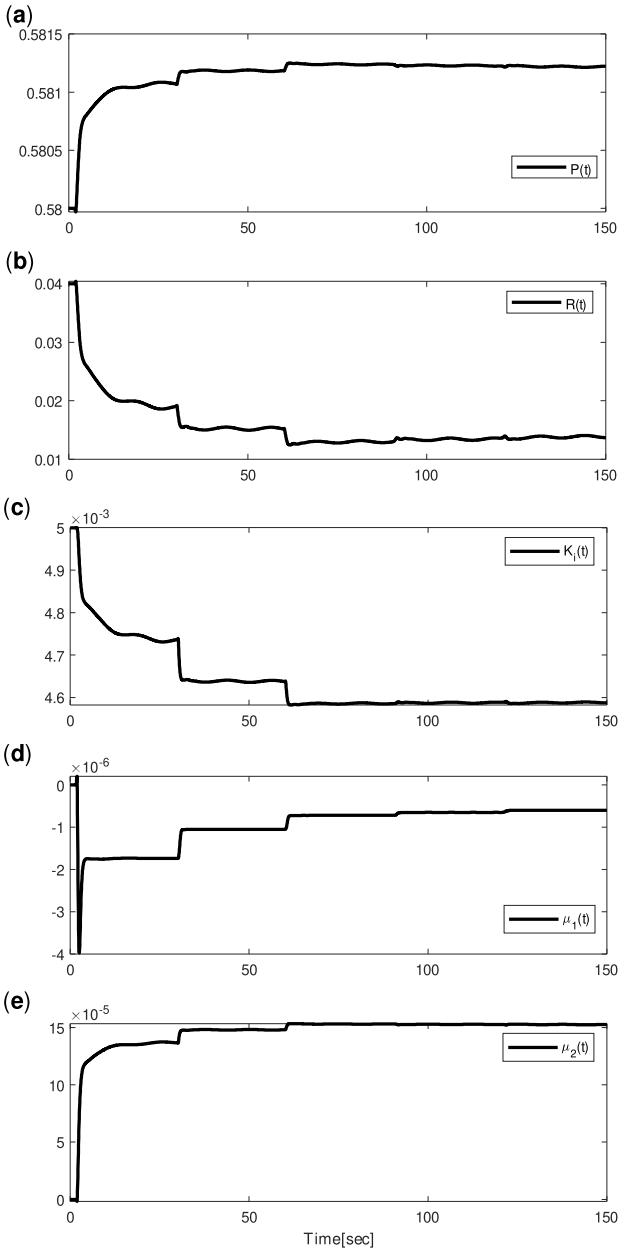


FIGURE 26. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System II (CSTR) (input disturbance case).

1) NOMINAL CASE

The trajectory tracking result obtained by the PID controller for benchmark system I is given In Fig(31) for the nominal case. The tracking performance can be seen in Fig. 31(a) and the control signal generated by the PID controller is depicted in Fig 31(b).

2) MEASUREMENT NOISE CASE

Fig. 32(a) illustrates the trajectory tracking performance obtained by the PID controller for the case when measurement noise is applied. Fig. 32(b) shows the control input signal computed by the PID controller. When we compare Fig.(32) with Fig. (10) it is clearly seen that the performance of the proposed method is better in dealing with measurement noise and the PID control signal fails to overcome the effects of the added noise.

3) INPUT DISTURBANCE CASE

In Fig.(33), the system output is given together with the control signal for the $u_{d_n} = 0.01\sin(\frac{2\pi}{12.5}n)$ input disturbance case when the PID controller is applied. When Fig.(33) and Fig. (13) are compared, it is clearly seen that the proposed method yields a far better performance than that obtained by the PID controller.

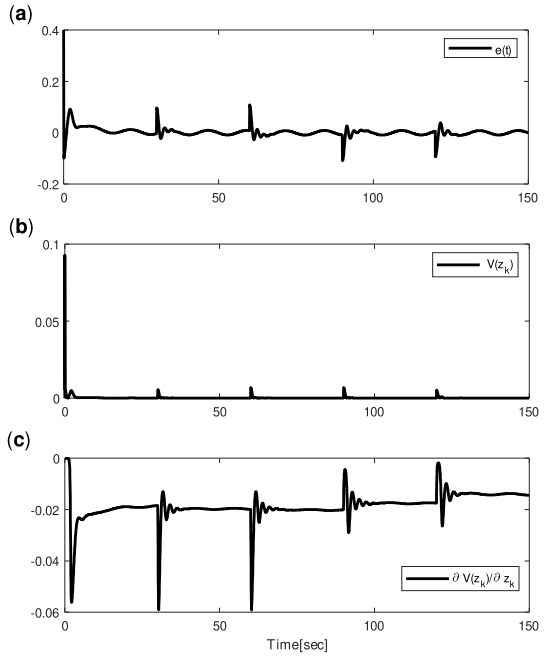


FIGURE 27. Trajectory Tracking error (a), Lyapunov Function (b), and its derivative (c), for Benchmark System II (CSTR) (input disturbance case).

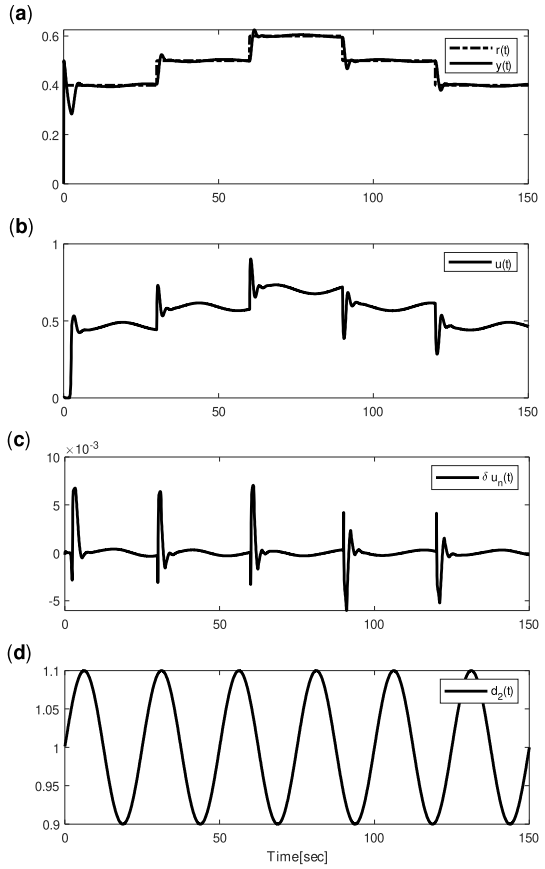


FIGURE 28. Tracking performance(a), Inverse optimal control signal (b), control signal correction term (c), and Uncertain system parameter (d) for Benchmark System II (CSTR) (parametric uncertainty case).

4) PARAMETRIC UNCERTAINTY CASE

Fig 34(a) depicts the trajectory tracking performance and Fig 34(b) graphs the control input signal computed by the

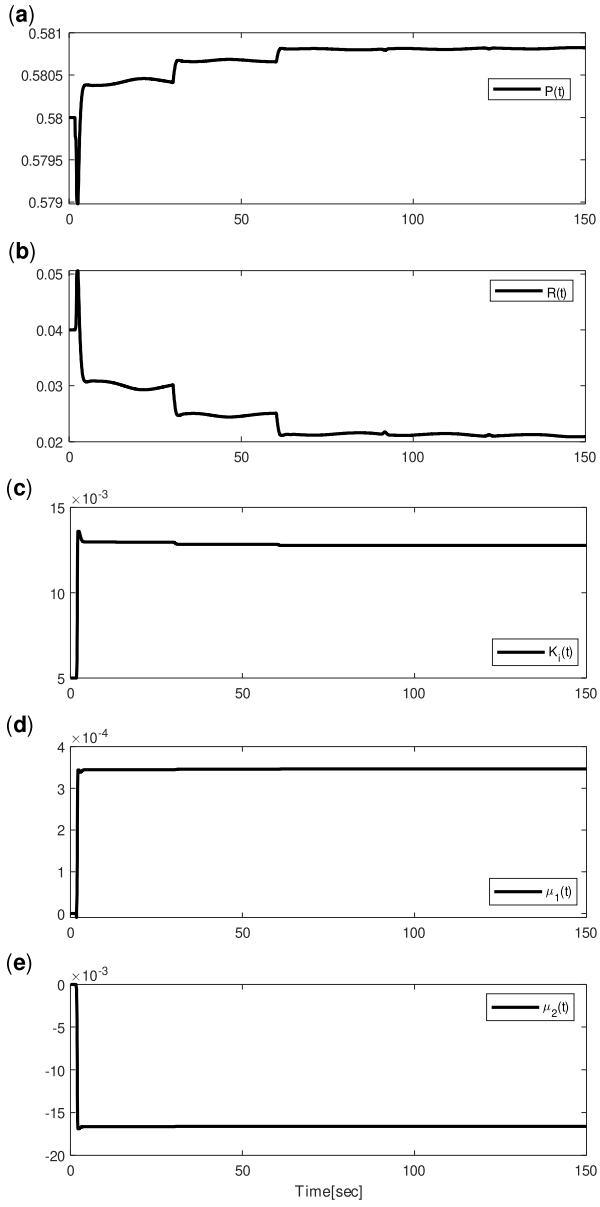


FIGURE 29. Adaptive P (a), R (b), and K_I (c) parameters, μ_1 values (d), μ_2 values (e) for Benchmark System II (CSTR) (parametric uncertainty case).

PID controller when a parametric uncertainty of $p_n = 0.1(1+0.1sin(2\pi\frac{1}{25}n))$ is applied. When figures (16) and (34) are compared, it is clearly seen that the proposed control method can successfully provide good tracking performance compared to PID controller and effectively deals with parametric uncertainty while PID totally fails.

B. BENCHMARK SYSTEM II

The ISE performance index values computed for the system given by (90) for all simulated cases are listed in Table 2. The table demonstrates the superiority of the proposed control method in comparison to the conventional PID controller for the nominal case and for the cases when measurement noise, input disturbance and parametric uncertainty are added to

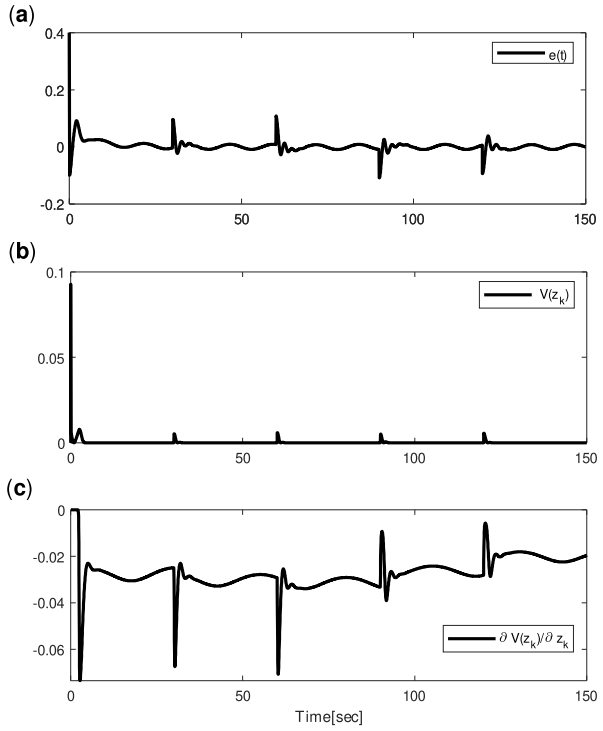


FIGURE 30. Trajectory Tracking Error (a), Lyapunov Function (b), and its derivative (c) for Benchmark System II (CSTR) (parametric uncertainty case).

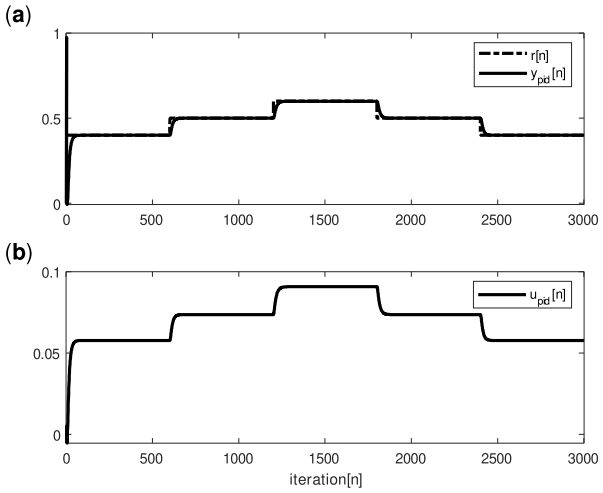


FIGURE 31. Tracking performance (a), PID control signal (b) for Benchmark System I (nominal case).

the system. The gain parameters of PID for all the cases are $K_p = 0.004$, $K_i = 0.041$, $K_d = 0.060$. As for the benchmark system I, the integral gain is selected the same as the value used for the proposed controller, the proportional and derivative gains are determined in accordance with the integral gain using Ziegler-Nichols design criteria.

1) NOMINAL CASE

Fig.(35) illustrates the trajectory tracking performance and the control input signal obtained by the PID controller for benchmark system II. When figures (19) and (35) are

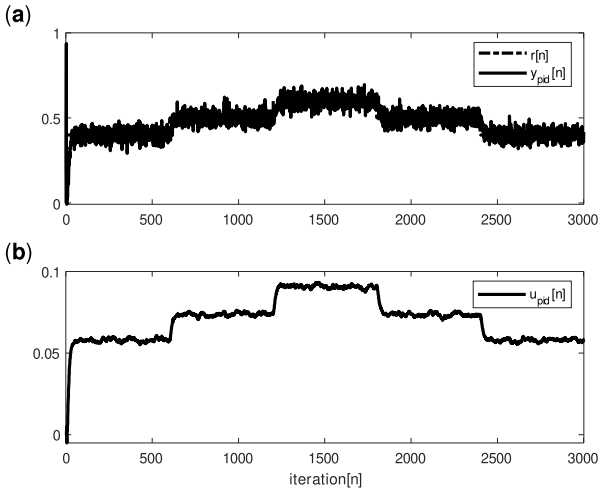


FIGURE 32. Tracking performance (a), PID control signal (b) for Benchmark System I (measurement noise case).

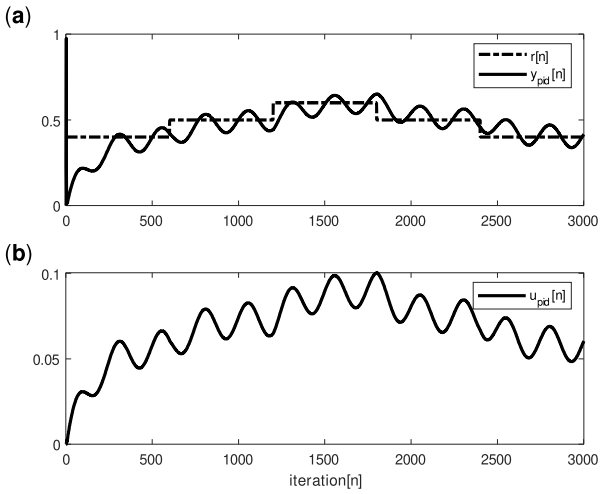


FIGURE 33. Tracking performance (a), PID control signal (b) for Benchmark System I (input disturbance case).

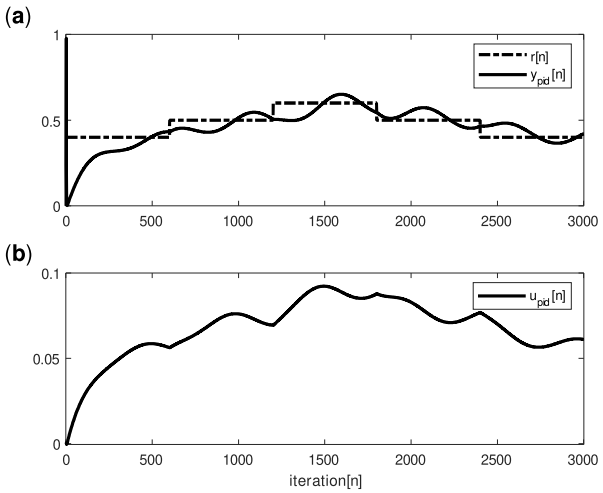


FIGURE 34. Tracking performance (a), PID control signal (b)) for Benchmark System I (parametric uncertainty case).

compared, it is easily seen that the proposed control methodology can provide successful tracking and the transient effects are eliminated compared to PID control.

TABLE 2. Comparison of proposed control method and PID controller with respect to ISE performance index(CSTR System in (83)).

Controller Type \ Cases	Nominal	Measurement Noise	Disturbance	Uncertainty
Inverse	0.0367	0.0603	0.0505	0.0490
PID	0.1353	0.2969	0.1506	0.1512

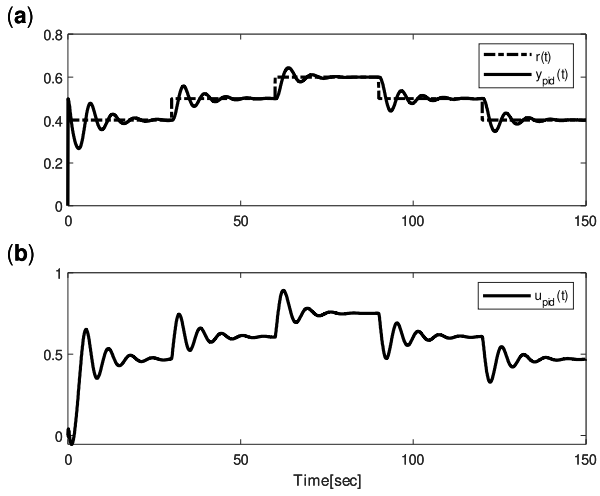


FIGURE 35. Tracking performance (a), PID control signal (b), and control signal correction term (c) for Benchmark System II (CSTR) (nominal case).

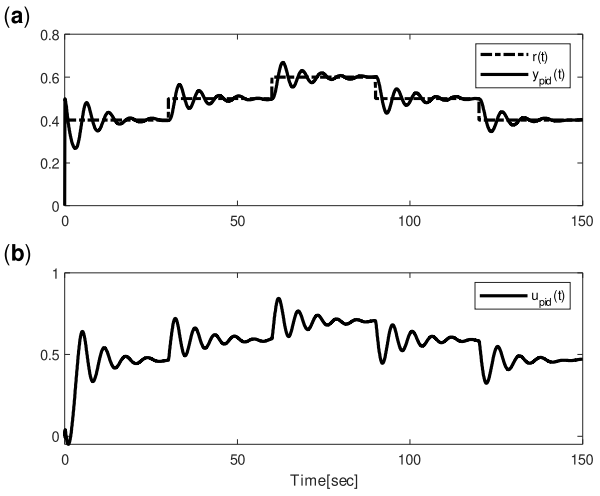


FIGURE 37. Tracking performance (a), PID control signal (b) for inputBenchmark System II (CSTR) (disturbance case).

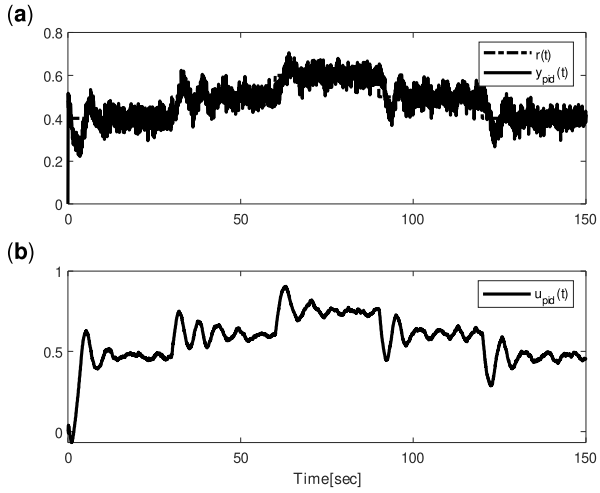


FIGURE 36. Tracking performance(a), PID control signal (b) for Benchmark System II (CSTR) (measurement noise case).

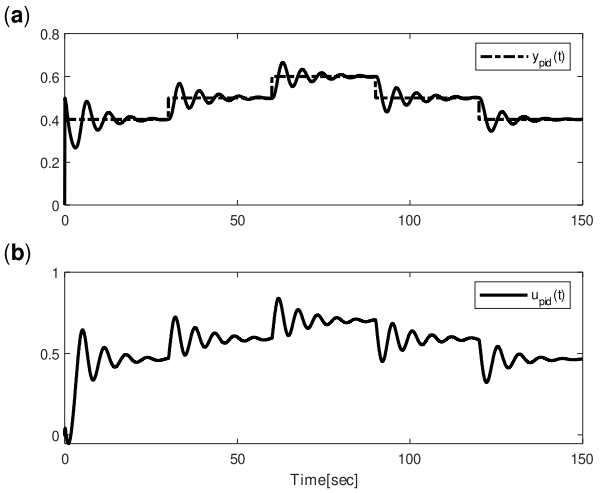


FIGURE 38. Tracking performance (a), PID control signal (b) for Benchmark System II (CSTR) (parametric uncertainty case).

2) MEASUREMENT NOISE CASE

Fig.(36) shows the trajectory tracking results obtained by the PID controller when a 30 dB measurement noise is applied to benchmark system II. When figures (22) and (36) are compared, it can be observed that the proposed control method is more successful compared to PID control. The control input computed by the PID controller fails to surpass measurement noise and transient effects.

3) INPUT DISTURBANCE CASE

In Fig.(37), tracking output and control signal obtained for the case when input disturbance of $d(t) = 0.01\sin(\frac{2\pi}{12.5}t)$ is applied to the benchmark system II is given for PID control. The results illustrated in figures (25) and (37) justify that in comparison to the PID control, the proposed control method can successfully provide good trajectory control and also it overcomes the effects of the disturbance.

4) PARAMETRIC UNCERTAINTY CASE

Fig.(38) illustrates the output trajectory and the control input attained by PID control when parametric uncertainty is added to the system. The comparison of figures (28) and (38) verifies that the obtained results by the proposed method are superior to those obtained by the PID controller.

VIII. CONCLUSION

In technical literature, the work on inverse optimal control mainly concentrates on application of the method on affine systems, however, the inverse optimal control method has not been implemented on non-affine systems yet. In this paper, a novel inverse optimal control approach is proposed for nonlinear and non-affine systems based on NARMA-L2 modelling and online LSSVR. First, the NARX model is obtained for non-affine systems using online LSSVR, then this model is decomposed into NARMA-L2 submodels. This procedure provides a conversion from non-affine to affine system model. Next, the computed NARMA-L2 submodels are used in the computation of the inverse optimal control law. Another contribution of the proposed method is that the parameters of the inverse optimal controller with integrator are all computed in an online manner using the Levenberg-Marquardt algorithm. Overall, the main novelties and contributions of this proposed work can be listed as, the application of inverse optimal control method to nonlinear non-affine systems by conversion from non-affine to affine system model using NARMA-L2 modeling method, utilization of online LSSVR in inverse optimal controller design and adaptive optimization of controller parameters of the inverse optimal controller in an online manner using Levenberg-Marquardt algorithm.

The resulting control architecture is an adaptive control system where the system model is estimated and all parameters are optimized online at every sampling time. The extensive simulations performed on benchmark problems to verify the success of the proposed control methodology.

For future studies, we firstly aim to extend the application of the inverse optimal control method to nonlinear non-affine systems by using different machine-learning based methods. Reinforcement based learning methods are good candidates for this goal. Also, it is within our plans to design novel adaptive inverse optimal controller architectures that do not require transformation parameters that enable the transition from NARX model to NARMA-L2 model. Furthermore, model predictive control concepts can be integrated with inverse optimal control methodology using machine-learning based methods. Another possible research direction is the implementation of inverse optimal control method for multi-input multi-output (MIMO) nonlinear non-affine systems.

DECLARATIONS

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] H. Shu and Y. Pi, "PID neural networks for time-delay systems," *Comput. Chem. Eng.*, vol. 24, nos. 2–7, pp. 859–862, Jul. 2000.

[2] M. E. Sanci, S. Halis, and Y. Kaplan, "Optimization of machining parameters to minimize surface roughness in the turning of carbon-filled and glass fiber-filled polytetrafluoroethylene," *Mater. Design Appl.*, vol. 65, pp. 295–305, Mar. 2017.

[3] M. O. Efe and O. Kaynak, "A comparative study of soft-computing methodologies in identification of robotic manipulators," *Robot. Auto. Syst.*, vol. 30, no. 3, pp. 221–230, Feb. 2000.

[4] M. O. Efe and O. Kaynak, "A comparative study of neural network structures in identification of nonlinear systems," *Mechatronics*, vol. 9, no. 3, pp. 287–300, Apr. 1999.

[5] M. A. Denai, F. Palis, and A. Zeghbib, "ANFIS based modelling and control of non-linear systems: A tutorial," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4, 2004, pp. 3433–3438.

[6] A. Gretton, A. Doucet, R. Herbrich, P. J. W. Rayner, and B. Scholkopf, "Support vector regression for black-box system identification," in *Proc. 11th IEEE Signal Process. Workshop Stat. Signal Process.*, Dec. 2001, pp. 341–344.

[7] H. Rong, G. Zhang, and C. Zhang, "Application of support vector machines to nonlinear system identification," in *Proc. Auto. Decentralized Syst. (ISADS)*, 2005, pp. 501–507.

[8] J. A. K. Suykens, "Nonlinear modelling and support vector machines," in *Proc. 18th IEEE Instrum. Meas. Technol. Conf. Rediscovering Meas. Age Informat. (IMTC).*, vol. 1, 2001, pp. 287–294.

[9] K. Uçak and G. Ö. Günel, "Generalized self-tuning regulator based on online support vector regression," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 775–801, Dec. 2017.

[10] K. S. Narendra and S. Mukhopadhyay, "Adaptive control using neural networks and approximate models," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 475–485, May 1997.

[11] M. Majstorovic, I. Nikolic, J. Radovic, and G. Kvascev, "Neural network control approach for a two-tank system," in *Proc. 9th Symp. Neural Netw. Appl. Electr. Eng.*, 2008, pp. 215–218.

[12] O. T. Nyandoro, J. O. Pedro, O. A. Dahunsi, and B. Dwolatzky, "Linear slip control formulation for vehicular anti-lock braking system with suspension effects," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 4779–4784, Jan. 2011.

[13] O. De Jesus, A. Pukrittayakamee, and M. T. Hagan, "A comparison of neural network control algorithms," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 1, 2001, pp. 521–526.

[14] A. Pukrittayakamee, O. De Jesus, and M. T. Hagan, "Smoothing the control action for NARMA-L2 controllers," in *Proc. 45th Midwest Symp. Circuits Syst. (MWSCAS)*, Sep. 2002, pp. 37–40.

[15] M. T. Hagan, H. B. Demuth, and O. De Jesús, "An introduction to the use of neural networks in control systems," *Int. J. Robust Nonlinear Control*, vol. 12, no. 11, pp. 959–985, Sep. 2002.

[16] Wahyudi, S. S. Mokri, and A. A. Shafie, "Real time implementation of NARMA L2 feedback linearization and smoothed NARMA L2 controls of a single link manipulator," in *Proc. Int. Conf. Comput. Commun. Eng.*, May 2008, pp. 691–697.

[17] A. Akbarimajd and S. Kia, "NARMA-L2 controller for 2-DoF underactuated planar manipulator," in *Proc. 11th Int. Conf. Control Autom. Robot. Vis.*, Dec. 2010, pp. 195–200.

[18] T. Vesselenyi, S. Dzitac, I. Dzitac, and M.-J. Manolescu, "Fuzzy and neural controllers for a pneumatic actuator," *Int. J. Comput. Commun. Control*, vol. 2, no. 4, pp. 375–387, Dec. 2007.

[19] P. A. Ioannou, *Adaptive Control of Systems with Actuator and Sensor Nonlinearities*, G. Tao and P. V. kokotovic, Eds. New York, NY, USA: Wiley, 1996.

[20] T. Chen, S. Zhang, and S. S. Ge, "UDE-based robust control for a class of nonaffine nonlinear systems with uncertainties and output constraint," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 247–252.

[21] Q. Zhang and C. Wang, "Robust adaptive backstepping control for a class of constrained non-affine nonlinear systems via self-organizing Hermite-polynomial-based neural network disturbance observer," *Adv. Mech. Eng.*, vol. 9, no. 5, May 2017, Art. no. 168781401770281.

[22] X. Bu, B. Jiang, and H. Lei, "Performance guaranteed finite-time non-affine control of waverider vehicles without function-approximation," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3252–3262, Mar. 2023.

[23] G. Li, Y. Liu, Y. Li, and X. Bu, "Adaptive back-stepping control of high-order uncertain nonlinear systems that a funnel control scheme with uncertain dynamics," in *Proc. Int. Conf. Electr. Eng. Control Technol. (CEECT)*, Dec. 2020, pp. 1–8.

[24] Y. Li, Q. Zhu, J. Zhang, and Z. Deng, "Adaptive fixed-time neural networks control for pure-feedback non-affine nonlinear systems with state constraints," *Entropy*, vol. 24, no. 5, p. 737, May 2022.

[25] T. D. Son and Q. Nguyen, "Safety-critical control for non-affine nonlinear systems with application on autonomous vehicle," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 7623–7628.

[26] S. Zhao, Y. Pan, P. Du, and H. Liang, "Adaptive control for non-affine nonlinear systems with input saturation and output dead zone," *Appl. Math. Comput.*, vol. 386, Dec. 2020, Art. no. 125506.

[27] A. Al-Tamimi, "Optimal controller design algorithm for non-affine in input discrete-time nonlinear system," *Jordan J. Mech. Ind. Eng.*, vol. 6, pp. 155–161, Apr. Apr. 2012.

[28] T. Zhao and S. Sui, "Adaptive control for a class of non-affine nonlinear systems via two-layer neural networks," in *Proc. 6th World Congr. Intell. Control Autom.*, vol. 1, 2006, pp. 958–962.

[29] H. Wang, Y. Tian, and C. Vasseur, "Non-affine nonlinear systems adaptive optimal trajectory tracking controller design and application," *Stud. Informat. Control*, vol. 24, no. 1, pp. 5–11, Mar. 2015.

[30] N. Lin, R. Chi, and B. Huang, "Data-driven set-point control for nonlinear nonaffine systems," *Inf. Sci.*, vol. 625, pp. 237–254, May 2023.

[31] T. Kim, G. Park, K. Kwak, J. Bae, and W. Lee, "Smooth model predictive path integral control without smoothing," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10406–10413, Oct. 2022.

[32] S. R. Nekoo and B. Geranmehr, "Nonlinear observer-based optimal control using the state-dependent Riccati equation for a class of non-affine control systems," *J. Control Eng. Appl. Informat.*, vol. 16, pp. 5–13, Jun. 2014.

[33] X. Zhang and Z. Hou, "Data-driven predictive point-to-point iterative learning control," *Neurocomputing*, vol. 518, pp. 431–439, Jan. 2023.

[34] M. Defoort, T.-M. Guerra, and A.-T. Nguyen, "Control of siso non-affine-in-control discrete-time systems using Takagi–Sugeno models," *IFAC-PapersOnLine*, vol. 52, no. 11, pp. 79–84, 2019.

[35] X. Bu, B. Jiang, and H. Lei, "Nonfragile quantitative prescribed performance control of waverider vehicles with actuator saturation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 3538–3548, Aug. 2022.

[36] X. Bu and Q. Qi, "Fuzzy optimal tracking control of hypersonic flight vehicles via single-network adaptive critic design," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 1, pp. 270–278, Jan. 2022.

[37] X. Bu, "Actor-critic reinforcement learning control of non-strict feedback nonaffine dynamic systems," *IEEE Access*, vol. 7, pp. 65569–65578, 2019.

[38] W. Bai, P. X. Liu, and H. Wang, "Neural-network-based adaptive fixed-time control for nonlinear multiagent non-affine systems," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, Aug. 2022.

[39] S. S. Ge, J. Zhang, and T. H. Lee, "Adaptive MNN control for a class of non-affine NARMAX systems with disturbances," *Syst. Control Lett.*, vol. 53, no. 1, pp. 1–12, Sep. 2004.

[40] P. Bagheri, L. Behjat, and Q. Sun, "Nonlinear control of a class of non-affine variable-speed variable-pitch wind turbines with radial-basis function neural networks," *ISA Trans.*, vol. 131, pp. 197–209, Dec. 2022.

[41] X. Bu, "Air-breathing hypersonic vehicles funnel control using neural approximation of non-affine dynamics," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2099–2108, Oct. 2018.

[42] X. Bu, C. Hua, M. Lv, and Z. Wu, "Flight control of waverider vehicles with fragility-avoidance prescribed performance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, pp. 5248–5261, Oct. 2023.

[43] R. Ruiz-Cruz, E. N. Sanchez, F. Ornelas-Tellez, A. G. Loukianov, and R. G. Harley, "Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1698–1709, Dec. 2013.

[44] L. Ulusoy, M. Güzelkaya, and İ. Eksin, "Fusion of inverse optimal and model predictive control strategies," *Trans. Inst. Meas. Control*, vol. 42, no. 6, pp. 1122–1134, Apr. 2020.

[45] R. Fotouhi and M. Pourgholi, "Discrete-time inverse optimal control for consensus of multi-agent systems via a novel meta-heuristic algorithm," in *Proc. 7th Int. Conf. Control, Instrum. Autom. (ICCIA)*, Feb. 2021, pp. 1–5.

[46] M. A. Perez-Villalpando, K. J. G. Tun, C. A. Arellano-Muro, and F. Fausto, "Inverse optimal control using metaheuristics of hydropower plant model via forecasting based on the feature engineering," *Energies*, vol. 14, no. 21, p. 7356, Nov. 2021.

[47] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 897–908, Aug. 2016.

[48] X. Cai, C. Lin, L. Liu, and X. Zhan, "Inverse optimal control for strict-feedforward nonlinear systems with input delays," *Int. J. Robust Nonlinear Control*, vol. 28, no. 8, pp. 2976–2995, May 2018.

[49] C. Villaseñor, J. Rios, N. Arana-Daniel, A. Alanis, C. Lopez-Franco, and E. Hernandez-Vargas, "Germinal center optimization applied to neural inverse optimal control for an all-terrain tracked robot," *Appl. Sci.*, vol. 8, no. 1, p. 31, Dec. 2017.

[50] B. S. Leon, A. Y. Alanis, E. N. Sanchez, F. Ornelas-Tellez, and E. Ruiz-Velazquez, "Neural inverse optimal control via passivity for subcutaneous blood glucose regulation in type 1 diabetes mellitus patients," *Intell. Automat. Soft Comput.*, vol. 20, pp. 279–295, Mar. 2014.

[51] C. López-Franco, M. López-Franco, A. Y. Alanis, J. Gómez-Avila, and N. Arana-Daniel, "Real-time inverse optimal neural control for image based visual servoing with nonholonomic mobile robots," *Math. Problems Eng.*, vol. 2015, May 2015, Art. no. 347410.

[52] Z. Lin, Z. Liu, Y. Zhang, and C. L. P. Chen, "Adaptive neural inverse optimal tracking control for uncertain multi-agent systems," *Inf. Sci.*, vol. 584, pp. 31–49, Jan. 2022.

[53] L. J. Ricalde and E. N. Sanchez, "Inverse optimal neural control of a class of nonlinear systems with constrained inputs for trajectory tracking," *Optim. Control Appl. Methods*, vol. 33, no. 2, pp. 176–198, Mar. 2012.

[54] V. G. Lopez, E. N. Sanchez, A. Y. Alanis, and J. D. Rios, "Real-time neural inverse optimal control for a linear induction motor," *Int. J. Control*, vol. 90, no. 4, pp. 800–812, Apr. 2017.

[55] J. D. Rios, A. Y. Alanis, M. Lopez-Franco, C. Lopez-Franco, and N. Arana-Daniel, "Real-time neural identification and inverse optimal control for a tracked robot," *Adv. Mech. Eng.*, vol. 9, no. 3, Mar. 2017, Art. no. 168781401769297.

[56] G. Hernandez-Mejia, A. Y. Alanis, and E. A. Hernandez-Vargas, "Neural inverse optimal control for discrete-time impulsive systems," *Neurocomputing*, vol. 314, pp. 101–108, Nov. 2018.

[57] K. J. Gurubel, E. N. Sanchez, A. Coronado-Mendoza, V. Zuniga-Grajeda, B. Sulbaran-Rangel, and L. Breton-Deval, "Inverse optimal neural control via passivity approach for nonlinear anaerobic bioprocesses with biofuels production," *Optim. Control Appl. Methods*, vol. 40, no. 5, pp. 848–858, Sep. 2019.

[58] T. B. Lopez-Garcia, E. N. Sanchez, and R. Ruiz-Cruz, "Real-time implementation of battery bank charge–discharge based on neural inverse optimal control," *IET Renew. Power Gener.*, vol. 13, no. 16, pp. 3124–3132, Dec. 2019.

[59] C. J. Vega, O. J. Suarez, E. N. Sanchez, G. Chen, S. Elvira-Ceja, and D. I. Rodriguez, "Trajectory tracking on uncertain complex networks via NN-based inverse optimal pinning control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 854–864, Mar. 2020.

[60] V. M. Chan, E. A. Hernández-Vargas, and E. N. Sánchez, "Neural inverse optimal control applied to design therapeutic options for patients with COVID-19," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–7.

[61] Y.-m. Li, X. Min, and S. Tong, "Adaptive fuzzy inverse optimal control for uncertain strict-feedback nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 10, pp. 2363–2374, Oct. 2020.

[62] Y. Li, X. Min, and S. Tong, "Observer-based fuzzy adaptive inverse optimal output feedback control for uncertain nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 6, pp. 1484–1495, Jun. 2021.

[63] X. Min, Y. Li, and S. Tong, "Adaptive fuzzy output feedback inverse optimal control for vehicle active suspension systems," *Neurocomputing*, vol. 403, pp. 257–267, Aug. 2020.

[64] Y. Sun, J. Xu, C. Chen, and W. Hu, "Reinforcement learning-based optimal tracking control for levitation system of Maglev vehicle with input time delay," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–13, 2022.

[65] E. Moulay and W. Perruquetti, "Stabilization of nonaffine systems: A constructive method for polynomial systems," *IEEE Trans. Autom. Control*, vol. 50, no. 4, pp. 520–526, Apr. 2005.

[66] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, "Nonlinear design of adaptive controllers for linear systems," *IEEE Trans. Autom. Control*, vol. 39, no. 4, pp. 738–752, Apr. 1994.

[67] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive, and Robust*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.

[68] S. S. Sastry and A. Isidori, "Adaptive control of linearizable systems," *IEEE Trans. Autom. Control*, vol. 34, no. 11, pp. 1123–1131, 1989.

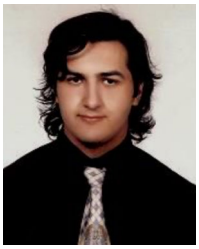
[69] Z. Artstein, "Stabilization with relaxed controls," in *Nonlinear Anal., Theory, Methods Appl.*, vol. 7, no. 11, pp. 1163–1173, Jan. 1983.

[70] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM J. Control Optim.*, vol. 21, no. 3, pp. 462–471, May 1983.

[71] P. S. P. da Silva and S. Batista, "On state representations of nonlinear implicit systems," *Int. J. Control*, vol. 83, no. 3, pp. 441–456, Mar. 2010.

[72] Y.-D. Song and Q. Song, "Survey of the latest developments in control of non-affine systems," in *Proc. 30th Chin. Control Conf.*, Jul. 2011, pp. 785–790.

[73] D. E. Kirk, *Optimal Control Theory: An Introduction* (Dover Books on Electrical Engineering Series). Mineola, NY, USA; Dover, 2004.
[74] R. Sepulchre, M. Jankovic, and P. V. Kokotovic," *Constructive Nonlinear Control*. London, U.K.: Springer, 2011.
[75] R. E. Kalman, "When is a linear control system optimal?" *J. Basic Eng.*, vol. 86, no. 1, pp. 51–60, Mar. 1964.
[76] R. A. Freeman and P. V. Kokotovic, "Inverse optimality in robust stabilization," *SIAM J. Control Optim.*, vol. 34, no. 4, pp. 1365–1391, Jul. 1996.
[77] B. S. Leon, A. Y. Alanis, E. N. Sanchez, F. Ornelas-Tellez, and E. Ruiz-Velazquez, "Neural inverse optimal control applied to type 1 diabetes mellitus patients," *Analog Integr. Circuits Signal Process.*, vol. 76, no. 3, pp. 343–352, Sep. 2012.
[78] E. N. Sanchez and F. Ornelas-Tellez," *Discrete-Time Inverse Optimal Control for Nonlinear Systems*. Boca Raton, FL, USA: CRC Press, Dec. 2017.
[79] B. Leon, V. Naumova, E. Ruiz-Velazquez, A. D. McCulloch, and E. Sanchez, "Combination of neural inverse optimal control with a kernel-based regularization learning algorithm to prevent hypoglycemia in type 1 diabetes patients," *IEEE Trans. Neural Netw. Learn. Syst.*, 2016.
[80] C. Atkinson and A. Osseiran, "Discrete-space time-fractional processes," *Fractional Calculus Appl. Anal.*, vol. 14, no. 2, pp. 201–232, Jun. 2011.
[81] C. E. Carrasco-Gutierrez and W. Sosa, "A discrete dynamical system and its applications," *Pesquisa Operacional*, vol. 39, no. 3, pp. 457–469, Dec. 2019.
[82] E. N. Sanchez, *Discrete-Time Recurrent Neural Control*. Boca Raton, FL, USA: CRC Press, Sep. 2018.
[83] O. Galor, *Discrete Dynamical Systems*. Berlin, Germany: Springer, 2007.
[84] E. N. Sanchez, *Discrete-Time Recurrent Neural Control: Analysis and Applications*. Boca Raton, FL, USA: CRC Press, 2018.
[85] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[86] J. W. Webb and R. A. Reis, *Programmable Logic Controllers: Principles and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
[87] K. Ucak and G. Oke, "An improved adaptive PID controller based on online LSSVR with multi RBF kernel tuning," in *Adaptive and Intelligent Systems* (Lecture Notes in Computer Science), vol. 6943. Berlin, Germany: Springer, 2011.
[88] K. Uçak and G. Ö. Günel, "A novel adaptive NARMA-L2 controller based on online support vector regression for nonlinear systems," *Neural Process. Lett.*, vol. 44, no. 3, pp. 857–886, Dec. 2016.
[89] K. Ucak and G. Öke, "RBF neural network controller based on OLSSVR," in *Proc. ICAIS*, in Lecture Notes in Computer Science, vol. 6943. Cham, Switzerland: Springer, 2011, pp. 40–51.
[90] G. D. Şen and G. Ö. Günel, "A NARMA-L2 controller based on online LSSVR for nonlinear systems," in *Proc. 15th Eur. Workshop Adv. Control Diagnosis (ACDE)*, E. Zattoni, S. Simani, and G. Conte, Eds. Cham, Switzerland: Springer, 2022, pp. 213–231.
[91] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
[92] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.
[93] S. R. Gunn, "Support vector machines for classification and regression," *ISIS Tech. Rep.*, vol. 14, no. 1, pp. 5–16, 1998.
[94] R. Middletone and G. Goodwin, "Adaptive computed torque control for rigid link manipulators," in *Proc. 25th IEEE Conf. Decis. Control*, Dec. 1986, pp. 68–73.
[95] G. D. Şen and G. Ö. Günel, "NARMA-L2–based online computed torque control for robotic manipulators," *Trans. Inst. Meas. Control*, vol. 45, no. 13, pp. 2446–2458, 2023.
[96] K. Uçak and G. Ö. Günel, "Online support vector regression based adaptive NARMA-L2 controller for nonlinear systems," *Neural Process. Lett.*, vol. 53, no. 1, pp. 405–428, Feb. 2021.
[97] S. Iplikci, "Controlling the experimental three-tank system via support vector machines," in *Proc. 9th Int. Conf. Adapt. Natural Comput. Algorithms (ICANNGA)*, Kuopio, Finland. Berlin, Germany: Springer, Apr. 2009, pp. 391–400.
[98] S. Wanfeng, Z. Shengdun, and S. Yajing, "Adaptive PID controller based on online LSSVM identification," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2008, pp. 694–698.
[99] Y.-F. Zhu and Z.-Y. Mao, "Online optimal modeling of LS-SVM based on time window," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, 2004, pp. 1325–1330.
[100] K. Uçak, "A novel model predictive Runge–Kutta neural network controller for nonlinear MIMO systems," *Neural Process. Lett.*, vol. 51, no. 2, pp. 1789–1833, Apr. 2020.
[101] S. Iplikci, "A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 20, no. 13, pp. 1483–1501, Sep. 2010.
[102] K.-Y. Show and D.-J. Lee, "Bioreactor and bioprocess design for biohydrogen production," in *Biohydrogen*, A. Pandey, J.-S. Chang, P. C. Hallenbecka, and C. Larroche, Eds. Amsterdam, The Netherlands: Elsevier, 2013, ch. 13, pp. 317–337.
[103] W. Wu and Y.-S. Chou, "Adaptive feedforward and feedback control of non-linear time-varying uncertain systems," *Int. J. Control*, vol. 72, no. 12, pp. 1127–1138, Jan. 1999.
[104] W. T. Miller, R. S. Sutton, and P. J. Werbos, *A Bioreactor Benchmark for Adaptive Network-Based Process Control*. Cambridge, MA, USA: MIT Press, 1995, pp. 387–402.
[105] K. Ogata, *Modern Control Engineering* (Instrumentation and controls series). Upper Saddle River, NJ, USA: Prentice-Hall, 2010.



MUHAMMET EMRE SANCI (Member, IEEE) received the B.S. degree in physics from Bolu Abant Izzet Baysal University, Bolu, Turkey, in 2011, and the M.S. degree in electrical and electronics engineering from Pamukkale University, Denizli, Turkey, in 2016. He is currently pursuing the Ph.D. degree in mechatronics engineering with Istanbul Technical University, Istanbul, Turkey. From 2013 to 2016, he was a Research Assistant with the Department of Mechatronics Engineering, Pamukkale University. Since 2017, he has been a Research Assistant with the Department of Control and Automation Engineering, Istanbul Technical University. His research interests include inverse optimal control, adaptive control, nonlinear non-affine systems, artificial intelligence, and machine learning.

KEMAL UÇAK (Member, IEEE) received the B.Sc. degree in electrical and electronics engineering from Pamukkale University, Denizli, Turkey, in 2007, and the M.Sc. and Ph.D. degree in control and automation engineering from Istanbul Technical University, Istanbul, Turkey, in 2012 and 2016, respectively.

He has been a Faculty Member of Istanbul Technical University and Muğla Sıtkı Koçman University, where he is currently an Assistant Professor with the Department of Electrical and Electronics Engineering. In 2020, he was a Postdoctoral Researcher with the Massachusetts Institute of Technology (MIT). His research interests include adaptive control, artificial intelligence, machine learning, model predictive control, nonlinear control, and robust adaptive control.

Dr. Uçak is a member of IFAC.

GÜLAY ÖKE GÜNEL received the M.S. and Ph.D. degrees from the Electrical and Electronics Engineering Department, Boğaziçi University, Turkey, in 1997 and 2004, respectively. In 2005 and 2009, she was a Postdoctoral Research Associate with the Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College London. In 2009, she joined as a Faculty Member of the Control and Automation Engineering Department, Istanbul Technical University, Turkey. Her research interests include neural networks, support vector machines, robotics, application of computational intelligence, and machine learning methods to the adaptive control of nonlinear systems.