



Neural Network Based Adaptive Inverse Optimal Control for Non-Affine Nonlinear Systems

Muhammet Emre Sancı¹ · Gülay Öke Günel¹

Accepted: 14 October 2023
© The Author(s) 2024

Abstract

In this paper, a novel methodology is introduced for the inverse optimal control of non-affine, nonlinear and discrete-time systems. Although inverse optimal control of affine systems is studied in detail in technical literature, there is no adequate research about its implementation on non-affine systems. here are two main contributions of this work. Firstly using the input–output data of the system to be controlled its NARMA-L2 model is obtained using a multi-layer feedforward neural network, this step provides a conversion from a non-affine to affine system model. After the affine system model is obtained, the inverse optimal control law is applied. The second contribution of this paper is the computation of the inverse optimal control signal. The selection of the P matrix in the control law is crucial since its value directly affects the control performance. Here a novel method is proposed where an adaptive and optimal P matrix is computed online using a recurrent neural network to minimize a predefined cost function. The performance of the proposed control method is evaluated by simulations performed on benchmark problems. The robustness of the method is also tested by additional simulations where noise and disturbance is imposed on the system. The obtained results justify the applicability of the proposed approach.

Keywords Adaptive control · Inverse optimal control · NARMA-L2 Model · Non-linear nonaffine systems

1 Introduction

Optimal control theory is a well-established area with an abundance of both theoretical and practical applications. The formulation of the optimal control law yields the Hamiltonian–Jacobi–Bellman (HJB) equation. Solution of the HJB equation for linear systems leads to the linear regulator problem and the optimal control law can be obtained using the Riccati equation. However, solving HJB equation for nonlinear systems is not a feasible task and

✉ Muhammet Emre Sancı
sanci17@itu.edu.tr

Gülay Öke Günel
gulayoke@itu.edu.tr

¹ Control and Automation Engineering, İstanbul Technical University, 34100 Maslak, İstanbul, Turkey

the solution cannot be easily computed. An alternative method for the optimal control of nonlinear systems is the inverse optimal control methodology.

The inverse optimal control (IOC) method has been implemented for the solution of numerous control problems. Numerical methods and machine learning algorithms have also been integrated with inverse optimal control.

Derivative-free global optimization methods have frequently been employed in order to optimize the parameters of the inverse optimal control method. A doubly fed induction generator (DFIG) was controlled by utilizing particle swarm optimization algorithm together with discrete-time inverse optimal control [1]. Model predictive control (MPC) method was merged with inverse optimal control using big bang-big crunch (BB-BC) optimization algorithm for estimation of a candidate control Lyapunov function [2]. Germinal center optimization algorithm (GCO) was employed to model a non-uniform competitive-based particle distribution to establish temporal leadership. This method was used in the design of a recurrent high order neural network (RHONN) to identify the model of the system to be controlled, then inverse optimal control law is designed with this identified model [3]. A linear quadratic regulator (LQR) has been designed based on Jaya algorithm (JA), teaching-learning algorithm (TLBO) and advanced teaching-learning (ATLBO) and the performance of these algorithms were compared in optimizing the cost function for a multi agent system with a focus on inverse optimal approach [4]. Inverse optimal control method with a recurrent neural network predictor structure has been applied to the control of a hydropower plant via adjustment of parameters by using the particle swarm optimization (PSO) algorithm [5].

Machine learning based methods have also been integrated with inverse optimal control technique. Inverse reinforcement learning (IRL) algorithm has been used to solve a tracking control problem where learning performance objective indices and control policies are based on inverse optimal control [6]. In another study, an inverse reinforcement learning algorithm was proposed which is able to infer the reward function without assuming a cooperative reward and instant communication for diverse set of agents. The proposed method was also extended to continuous inverse optimal control [7]. Additionally, fuzzy/neural-based adaptive optimal control strategies for continuous and discrete time nonlinear affine systems have also been developed. Fuzzy logic systems have been utilized for the investigation of the adaptive fuzzy inverse optimal control problem for a class of uncertain strict-feedback nonlinear systems [8]. The immeasurable states in nonlinear uncertain systems were identified by an observer-based fuzzy adaptive inverse optimal output feedback controller design [9]. For the quarter-car active suspension system (ASS) model, an adaptive fuzzy output feedback inverse optimal controller has been applied [10].

Neural network based methods have also been implemented with inverse optimal control methodology. For trajectory tracking of nonlinear systems with constrained inputs a recurrent higher order neural network (RHONN) was used in the inverse optimal control design [11]. RHONN structure has been integrated with inverse optimal control in many diverse applications [1–4, 12–20]. For trajectory tracking problem of uncertain complex networks, a neural network based inverse optimal pinning control strategy has been proposed [21]. A neural identifier based on RHONN was used to identify the dynamics of a robot and inverse optimal control was applied for its tracking control [22]. In another study, 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 in order to prevent hyperglycemia [23]. An inverse optimal neural controller was utilized also in visual feedback control of mobile robots with non-holonomic constraints [24]. In an attempt to apply inverse optimal control to data obtained via motion capture of collaborative manipulation in a shared workspace, human-robot collaboration was actualized to learn a cost function that

predicts how humans move [25]. RHONN was employed to obtain the mathematical model of a linear induction motor with uncertainties and inverse optimal control was applied for its control [26]. RHONN was also applied for the identification of impulsive systems and using the obtained model, impulsive neuro-control method was proposed using inverse optimal control technique [27]. Based on the neural model obtained under the presence of unknown bounded disturbances and parameter uncertainties, an inverse optimal neural control method has been developed by using the passivity concept. The proposed method was used to stabilize a bio-fuel production process [28]. In another application of RHONN, it was employed for online identification for both charge and discharge processes of a battery bank. A supervised control method was developed based on inverse optimal control to solve the reference tracking problem of the battery bank [29]. There are several other applications of RHONN integrated with inverse optimal control method. Neural inverse optimal control approach has been developed to improve the low-voltage ride-through capacity for a grid connected doubly fed induction generator. In this work, RHONN was utilized to identify the doubly fed induction generator and DC-link dynamics [30]. By using a RHONN identifier, a neural inverse optimal control algorithm based controller has been designed to represent the viral dynamics of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) patients and schedule theoretical therapies for curing the disease [31]. Neural network architectures other than RHONN have also been utilized with inverse optimal control. Neural networks have been employed in prediction, an inverse optimal control based predictor has been designed for strict-feedforward systems with input delays [32]. Additionally, linearizable nonlinear systems with input delays have been converted to linear systems based on inverse optimal predictors by using a differential game approach [33]. In another work, neural networks have been utilized based on the backstepping technique in the controller with the aim of identifying the unknown dynamics. An adaptive neural inverse optimal based consensus control method which minimizes a meaningful cost function has also been proposed [33].

In all of these examples from the technical literature, inverse optimal control methodology was 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, the inverse optimal control technique has not been extensively studied for non-affine systems. In this paper, we propose a method for non-affine nonlinear systems, where we first convert the non-affine system model to an affine model, by using NARMA-L2 modelling technique. This makes it possible for us to utilize the inverse optimal control formulation derived for affine nonlinear systems. The NARMA-L2 model is obtained by using a feedforward neural network. Next, a control Lyapunov function is constructed by employing a recurrent neural network. The neural network is utilized online to compute the Lyapunov function continuously, hence the inverse optimal control method is implemented in an adaptive control architecture.

In a nutshell, the novel contributions of this paper are:

- A neural network based inverse optimal control method is proposed for non-affine nonlinear systems, where NARMA-L2 modelling technique is used to convert from the original non-affine system model to affine system model. A feedforward neural network is utilized in obtaining the NARMA-L2 model.
- The P matrix in the control Lyapunov function is updated continuously using a recurrent neural network.
- An adaptive control architecture is implemented for inverse optimal control method where the optimal values of the P matrix components are computed online using a neural network.

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 stabilization and trajectory tracking control. Moreover, robustness of the method is also justified by the simulation results obtained under disturbance and noise.

After this brief introduction, Sect. 2 summarizes the principles of inverse optimal control method. Sections 3 and 4 recap the essentials of affine and non-affine systems and NARMA-L2 models respectively. The proposed method is given in detail in Sect. 5. Simulation results are presented in Sect. 6. The paper ends with a brief conclusion in Sect. 7.

2 Inverse Optimal Control

Inverse optimal control is an approach where 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. Essential features of inverse optimal control are described below [17, 18, 34].

Consider a nonlinear, discrete and affine system given as [20, 35, 36]:

$$x_{n+1} = f(x_n) + g(x_n)u_n \quad (1)$$

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

For the system given in Eq. (1) it is desired to find a control law which minimizes the following control Lyapunov functional:

$$V(x_n) = \sum_{i=n}^{\infty} l(z_n) + u_n^T Q u_n \quad (2)$$

In Eq. (2), $V : \mathbb{R}^n \rightarrow \mathbb{R}^+$ represents the Lyapunov function, $l : \mathbb{R}^n \rightarrow \mathbb{R}^+$ is a positive semi-definite function and $Q : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ denotes a real symmetric and positive definite weighting matrix. We can rewrite Eq. (2) as:

$$\begin{aligned} V(x_n) &= l(x_n) + u_n^T R u_n + \sum_{i=n+1}^{\infty} l(x_n) + u_n^T R u_n \\ &= l(x_n) + u_n^T R u_n + V(x_{n+1}). \end{aligned} \quad (3)$$

From Bellman's optimality principle [37], it is clear that the value function $V^*(x_n)$ becomes time-invariant and satisfies the discrete-time Bellman equation for the infinite horizon optimization case [37].

$$V^*(x_n) = \min_{u_n} \left\{ l(x_n) + u_n^T R u_n + V^*(x_{n+1}) \right\}. \quad (4)$$

It must be noted that the Bellman equation is resolved backward in time. In order to determine the requirements that the optimal control law must meet, we define the discrete-time Hamiltonian $\mathcal{H}(x_n, u_n)$ as:

$$\mathcal{H}(x_n, u_n) = l(x_n) + u_n^T R u_n + V^*(x_{n+1}) - V^*(x_n) \quad (5)$$

This is used to obtain the control law u_n by calculating

$$\min_{u_n} \mathcal{H}(x_n, u_n) \tag{6}$$

The value of u_n , which achieves this minimization, is a feedback control law denoted as $u_n = \bar{u}(x_n)$, then

$$\min_{u_n} \mathcal{H}(x_n, u_n) = \mathcal{H}(x_n, \bar{u}(x_n)) \tag{7}$$

A necessary condition, which the feedback optimal control law $u(x_n)$ must satisfy [38] is:

$$\mathcal{H}(x_n, \bar{u}_n) = 0 \tag{8}$$

$u(x_n)$ is obtained by computing the gradient of the right-hand side of Eq. (5) with respect to u_n and equating it to zero.

$$\begin{aligned} 0 &= 2Ru_n + \frac{\partial V^*(x_{n+1})}{\partial u_n} \\ &= 2Ru_n + g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}}. \end{aligned} \tag{9}$$

The optimal control law is formulated as:

$$\begin{aligned} u_n^* &= \bar{u}(x_n) \\ &= -\frac{1}{2}R^{-1}g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}}, \end{aligned} \tag{10}$$

Here $u(0) = 0$ and $u(x_n)$ is a state feedback control law. Hence, $V(x_n)$ becomes a Lyapunov function and the boundary condition $V(0) = 0$ in Eqs. (2) and (3) is satisfied. The notation u_n^* is used to stress that u_n^* is the optimal solution.

Moreover, if $\mathcal{H}(x_n, u_n)$ is quadratic in u_n and $R > 0$, then the inequality:

$$\frac{\partial^2 \mathcal{H}(x_n, u_n)}{\partial u_n^2} > 0 \tag{11}$$

holds as a necessary condition such that the optimal control law given in (10) minimizes the performance index given in (2). When (10) is substituted in (4), we get

$$\begin{aligned} V^*(x_n) &= l(x_n) + \left(-\frac{1}{2}R^{-1}g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}}\right)^T \\ &\quad \times R \left(-\frac{1}{2}R^{-1}g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}}\right) + V^*(x_{n+1}) \\ &= l(x_n) + V^*(x_{n+1}) + \frac{1}{4} \frac{\partial V^{*T}(x_{n+1})}{\partial x_{n+1}} g(x_n) R^{-1} g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}} \end{aligned} \tag{12}$$

which can be rewritten as

$$l(x_n) + V^*(x_{n+1}) - V^*(x_n) + \frac{1}{4} \frac{\partial V^{*T}(x_{n+1})}{\partial x_{n+1}} g(x_n) R^{-1} g^T(x_n) \frac{\partial V^*(x_{n+1})}{\partial x_{n+1}} = 0 \tag{13}$$

Equation (13) is known as the discrete-time HJB equation [37]. Solving this partial differential equation for $V^*(x_n)$ is not straightforward. This is one of the main drawbacks in discrete-time optimal control for nonlinear systems. To overcome this problem, inverse optimal control method has been proposed.

Definition 1 (*Radially unbounded function*) A positive definite function $V(x_n)$ satisfying $V(x_n) \rightarrow \infty$ as $\|x_n\| \rightarrow \infty$ is said to be radially unbounded.

Definition 2 (*Control Lyapunov function*) Let us make the assumption 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 are 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 $V(\cdot)$ is said to be a discrete-time control Lyapunov function (CLF) for (1).

Theorem 1 (*Global asymptotic stability*) The equilibrium point $x_n = 0$ of (1) is globally asymptotically stable if there exists a function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that

- (i) V is a positive definite function, decrescent and radially unbounded.
- (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 2 [39]

$$u_n^* = -\frac{1}{2}R^{-1}g^T(x_n)\frac{\partial V(x_{n+1})}{\partial x_{n+1}} \tag{14}$$

is inverse optimal if

- The equilibrium point $x_n = 0$ achieves exponential (global) stability for system (1).
- It minimizes a cost functional defined at (2) for

$$\bar{V} := V(x_{(n+1)}) - V(x_n) + u_n^{*T} R u_n^* \leq 0 \tag{15}$$

where the boundary condition is $V(0) = 0$.

In the practical implementation of the inverse optimal control method for stabilization control, a CLF satisfying the required properties [39] is selected as:

$$V(x_n) = \frac{1}{2}x_n^T P x_n \quad P = P^T \succ 0 \tag{16}$$

where P a symmetric positive definite matrix, i.e., $P = P^T \succ 0$ Then a system given by (1) with the control law given u_n^* :

$$u_n^* = -\frac{1}{2}(R + P_1(x_n))^{-1}P_2(x_n) \tag{17}$$

where $P_1(x_n) = g^T(x_n)P f(x_n)$ and $P_2(x_n) = \frac{1}{2}g^T(x_n)P g(x_n)$, ensures stability.

For a detailed proof of the Theorem 2, refer to [39].

Theorem 3 [23] Define the tracking error for x_n as $z_n = x_n - x_{\delta,n}$ where $x_{\delta,n}$ is the desired trajectory of x_n .

The cost functional given in Eq. (2) must be minimized. We need the boundary condition $V(0) = 0$ so $V(z_n)$ becomes a Lyapunov function. The optimal control law to achieve trajectory tracking is formulated as:

$$u_{c_n} = -\frac{1}{2}R^{-1}g^T(x_n)\frac{\partial V(z_{n+1})}{\partial z_{n+1}} \tag{18}$$

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

$$l(z_n) + V(z_{n+1}) - V(z_n)$$

$$+ \frac{1}{4} \frac{\partial V^T(z_{n+1})}{\partial z_{n+1}} g^T(x_n) R^{-1}(z_n) g(x_n) \frac{V(z_{n+1})}{\partial z_{n+1}} = 0 \tag{19}$$

Equation (18) is an inverse optimal (globally) stabilizing control input along $x_{\delta,n}$ if the following two conditions are satisfied:

Condition (1) Equation (18) provides (global) asymptotic stability of $x_n = 0$ for the system given by (1) along $x_{\delta,n}$.

Condition (2) The positive definite and radially unbounded CLF $V(z_n)$ satisfies:

$$\bar{V} := V(z_{n+1}) - V(z_n) + u_{c_n}^T R(z_n) u_{c_n} \tag{20}$$

If $l(z_n) := -\bar{V}$, then $V(z_n)$ is a solution for Eq. (19) and it follows that the cost functional of the tracking error is minimized.

In the practical implementation of the inverse optimal control method for trajectory tracking control problem, a CLF satisfying the necessary properties [23] is selected as:

$$V(z_n) = \frac{1}{2} z_n^T P z_n \quad P = P^T \succ 0 \tag{21}$$

with the tracking error (z_n):

$$z_n = x_n - x_{\delta,n} = \begin{bmatrix} (x_{1,n} - x_{1\delta,n}) \\ \vdots \\ \vdots \\ (x_{n,n} - x_{n\delta,n}) \end{bmatrix} \tag{22}$$

The P parameter must be selected as symmetric positive definite. This is the minimum condition that P must satisfy. After that it can be optimized to minimize the stabilization or tracking error. Various methods have been proposed for the optimization and this is one of the main focuses of the research on inverse optimal control. In our paper we proposed a novel method where we optimize the P parameter adaptively with a recurrent neural network.

The inverse optimal control law can be formulated as:

$$\begin{aligned} u_{c_n} &= \left| -\frac{1}{4} R^{-1} g^T(x_n) \frac{\partial z_{n+1}^T P z_{n+1}}{\partial z_{n+1}} \right| \\ &= \left| -\frac{1}{2} (R + P_2(x_n))^{-1} P_1(x_n, x_{\delta,n}) \right| \end{aligned} \tag{23}$$

where

$$P_1(x_n, x_{\delta,n}) = \begin{cases} g^T(x_n) P (f(x_n) - x_{\delta,n+1}) & \text{for } f(x_n) \geq x_{\delta,n+1} \\ g^T(x_n) P (x_{\delta,n+1} - f(x_n)) & \text{for } f(x_n) \leq x_{\delta,n+1} \end{cases} \tag{24}$$

and

$$P_2(x_n) = \frac{1}{2} g^T(x_n) P g(x_n) \tag{25}$$

The control input given in (23) is the inverse optimal control law and it ensures the minimization of the cost function given in (21).

The detailed proof of Theorem 3 can be found in [23, 36, 39].

By the given theorem above, the inverse optimal control law can be computed with selection of appropriate matrix P for discrete time affine-in-input nonlinear system models.

3 Non-Affine Nonlinear Systems

A nonlinear and affine continuous system is represented as:

$$\dot{x} = f(x) + g(x)u \quad (26)$$

Here $x \in R^n$ is the vector of system states, $u \in R^m$ is the control input signal vector, $f(\cdot)$ and $g(\cdot)$ are nonlinear and smooth functions. Note that in this representation, the control input signal appears linearly and this provides practicability in implementation. However most of the physical systems cannot be modelled using the affinity assumption since they are inherently nonlinear. The most general formulation for nonlinear systems is:

$$\dot{x} = f(x, u) \quad (27)$$

When $f(x, u)$ is a smooth function, Eq. (27) can be rewritten in Taylor series expansion form as in (28) [40]:

$$\dot{x} = f_0(x) + \sum_{j=1}^k f_j(x)u^{[j]} + R(x, u) \quad (28)$$

In the Taylor series expansion, the terms higher than the first order terms can be neglected, which suggests the use of affine system model to approximate nonlinear systems. The common assumptions in modelling are that the plant under study is affine, i.e., model is linear in the input variables and the nonlinearities are linearly parameterized by unknown parameters. Nevertheless, many practical systems, e.g., chemical reactions, PH neutralization processes, etc., are inherently nonlinear, whose input variables may not be expressed in an affine form.

Feedback linearization method has frequently been applied for non-linear systems [41–43]. The prominent issue in control of non-linear systems is stabilization. According to Artstein's theorem [44], if a CLF exists for affine systems, this means that continuous feedback control also exists. For affine systems a general formulation for feedback law using CLF is provided in [45]. However for non-affine systems other formulations are provided for stabilization [40].

In discrete-time, the representations of non-affine and affine systems are given respectively as:

$$x_{(n+1)} = f(x_n, u_n) \quad (29)$$

$$x_{(n+1)} = f(x_n) + g(x_n)u_n \quad (30)$$

In this work, a technique to apply inverse optimal control method to non-affine systems is proposed. This method is based on derivation of the NARMA-L2 model of the system to be controlled and hence converting from a non-affine to affine system model.

4 NARMA-L2 System Identification

In the adaptive control of non-linear systems with unknown model, the first step is the identification of the system. In other words, a model that relates the inputs and outputs of the system with minimum error must be obtained. Design of controllers in general depends on the correct identification of the system model, so the system model should be precisely obtained for a good control performance. A discrete-time nonlinear and non-affine system

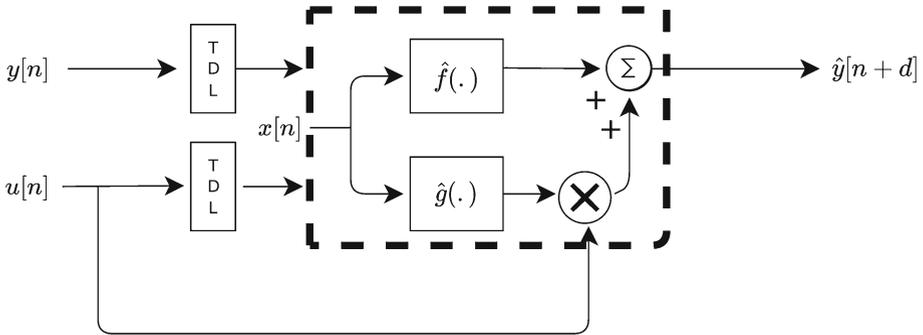


Fig. 1 A schematic diagram for NARMA-L2 model

can generally be represented by:

$$y[n + d] = F_{actual}(u[n], \dots, [n - k + 1], y[n], \dots, y[n - k + 1]) \tag{31}$$

where $y[n]$ is the system output, $u[n]$ represents the control input and d is relative degree. If the Taylor series expansion of (31) is written down and only the first order terms are considered, NARMA-L2 model is obtained. Hence, in the neighborhood of an equilibrium state NARMA-L2 model can be expressed as:

$$\hat{y}[n + d] = \hat{f}(x[n]) + \hat{g}(x[n])u[n] \tag{32}$$

Here, $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ are two nonlinear functions to be approximated and $x[n] = [u[n - 1] \cdots u[n - n_u], y[n], \dots, y[n - n_y + 1]]^T$ is the input vector for the estimators. n_u and n_y symbolize the instances of past control inputs and system outputs included in the input vector. It is clearly seen from (32) that the control input $u[n]$ appears linearly, separated from the nonlinear dynamics. So, the NARMA-L2 model represents an affine system. An illustration of the NARMA-L2 model is given in Fig. 1.

The main advantages of the NARMA-L2 model are its simplicity in implementation and its requirement of only two submodels to be estimated. Also, in NARMA-L2 model, the control signal appears linearly, thus it yields an affine model, which is easier to deal with. There are various control applications based on the NARMA-L2 model in the literature [46–50].

Up to now, the work on inverse optimal control method based on CLF in the literature mainly concentrates on affine systems. In this study, we propose a method to obtain the inverse optimal control law for non-affine systems, by making use of the NARMA-L2 model of the plant to be controlled. Hence, given the input–output data for the non-affine NARX model of the system, the NARMA-L2 model is computed, and using this model the CLF-based inverse optimal control law can be obtained. For this purpose, $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ submodels must be estimated. It is essential that these submodels are precisely estimated since they are utilized in the computation of the inverse optimal control law given in (23). In this study, we have employed a feedforward multilayer neural network to estimate $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ submodels, the neural network parameters are trained in order to minimize the error between system and model outputs $[(e(n + d) = y[n + d] - \hat{y}[n + d])]$. After identifying the dynamics of the system as in the model given in Eq. (1), the obtained submodels are used to compute the inverse optimal control law.

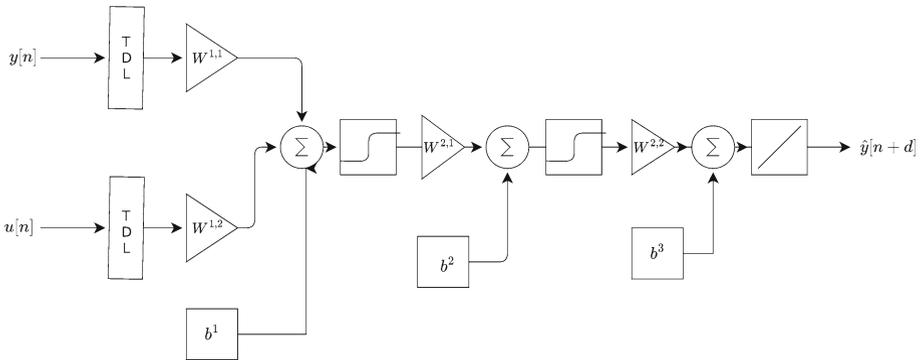


Fig. 2 NARMA-L2 neural network structure for system identification

5 Neural Network Based Adaptive Inverse Optimal Controller For Non-Affine Systems

In the technical literature, there are numerous implementations of inverse optimal control methodology for affine systems, where the control input appears linearly in the model [51–56]. However, there is no research on application of inverse optimal control method to non-affine systems. In this paper, a novel approach for the inverse optimal control of non-affine nonlinear systems is proposed. This method is based on converting from a NARX model to a NARMA-L2 model, hence an originally non-affine system can be represented as an affine model. The overall method consists of two steps. In the first step, the input–output data of the plant to be controlled is derived, then a feedforward neural network is utilized to obtain the NARMA-L2 model of the plant from the obtained dataset. When the input–output data is collected from the plant, this data represents the NARX model of the system, which is a non-affine model. By converting from NARX model to NARMA-L2 model, an affine model is attained from a non-affine model. The schematic diagram for the feedforward neural network used in converting from the NARX model to NARMA-L2 model is given below (Fig. 2):

The inputs to the neural network are the input–output data of the plant to be modelled. The input and output data are multiplied by different sets of weights in the first layer, they are fed into the activation function after a bias value is added. The neural network consists of four layers, in the first, second and third layers hyperbolic tangent function is used and in the last layer linear function has been utilized. The output of the neural network is the submodel of the NARMA-L2 model, separate neural networks are employed to estimate $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$ separately. This identification step for the NARMA-L2 model is carried out offline. After the identification of the plant to be controlled is completed, the second stage is the adaptive online implementation of the inverse optimal control methodology. The formulation of the inverse optimal control law has been given in Eqs. (18) and (17), in Sect. 2. In this equation the P matrix is critical, its value has a direct effect on the control performance. Hence various methods have been proposed in the literature for the optimization of the P matrix [57–59].

In this paper, a novel approach is proposed where P matrix is determined optimally in an adaptive manner. A recurrent neural network which works online computes the optimal values of the components of the P matrix continuously, so that a metric which is a function of the control error is minimized. The schematic diagram of the proposed control architecture is illustrated in Fig. 3.

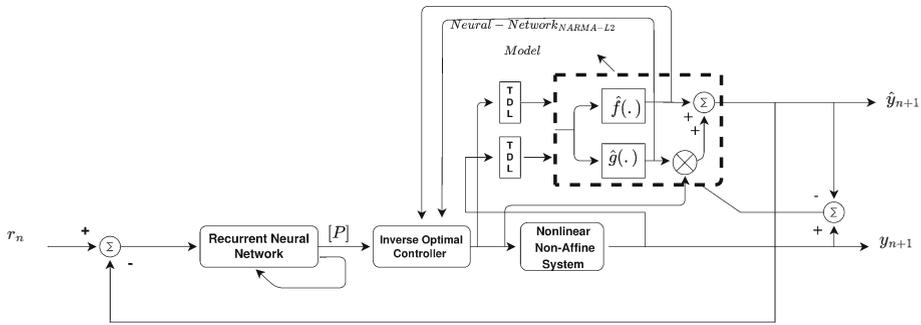


Fig. 3 Proposed control structure

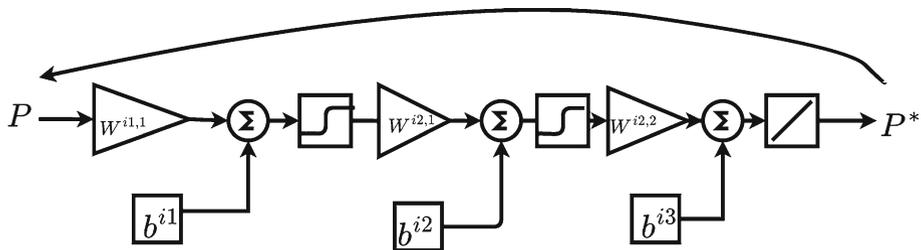


Fig. 4 Recurrent neural network structure for the calculation of controller parameters

In this architecture the NARMA-L2 sub-blocks $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$, which have been identified with the feedforward neural network, are used to compute the estimated system output $y[n + d]$. In order to compute the P matrix entries, a recurrent neural network is employed as depicted above (Fig. 4):

The optimal P^* matrix is computed by the recurrent neural network as:

$$P^* = W^i(n)^T P W^i(n) \tag{33}$$

The weights are updated so that a discrete Lyapunov function is minimized.

For stabilization, the Lyapunov function is defined as:

$$V^*(x_n) = \frac{1}{2} x_n^T P^* x_n \tag{34}$$

For trajectory tracking, it is formulated as:

$$V^*(z_n) = \frac{1}{2} z_n^T P^* z_n \tag{35}$$

For trajectory tracking problem, an error is calculated as the difference between the reference signal and the estimated system output:

$$e_t[n] = r[n] - \hat{y}[n + d] \tag{36}$$

For stabilization problem, error is obtained as the change in Lyapunov function:

$$e_s[n] = V(x_{(n+1)})^* - V(x_n)^* \tag{37}$$

If there exists a control Lyapunov function, then the system given in Eq. (1) is stabilizable. Thus we may utilize the Lyapunov function as the error to the Recurrent Neural Network for the calculation of the P matrix of the inverse optimal controller.

Hence, for trajectory tracking the weights are updated according to:

$$W^i(n + 1) = W^i(n) - \alpha P \sum_{k=1}^N \left| \frac{e_t}{n} \right| \tag{38}$$

For stabilization, weights are updated as:

$$W^i(n + 1) = W^i(n) - \alpha P \sum_{k=1}^N \left| \frac{e_s}{n} \right| \tag{39}$$

Consequently, the elements of the P matrix are updated.

The step by step procedure of the computation of the optimal P^* matrix by the recurrent neural network is given in the following:

STEP1—OFFLINE SYSTEM IDENTIFICATION

1. Initialize controller and model parameters:
 - P, R, W^i, W, n_u, n_y .
 - Generate a control input signal $u(n)$. Apply this signal to the plant to be controlled to compute system output $y(n)$.
2. – Obtain a training dataset by forming $[\mathbf{u}(\mathbf{n}) \mathbf{y}(\mathbf{n})]^T$ by using the input–output pairs generated in Step 1.
 - Here

$$\begin{aligned} \mathbf{u}(\mathbf{t}) &= [u(n) \ u(n - 1)u(n - 2) \ u(n - 3)] \\ \mathbf{y}(\mathbf{t}) &= [y(n) \ y(n - 1)y(n - 2) \ y(n - 3)] \end{aligned} \tag{40}$$

3. Construct feedforward neural networks as described in detail above and train them using the dataset constructed in Step 2 to estimate NARMA-L2 submodels $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$. Separate feedforward neural networks are employed to estimate $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$.

STEP2—ADAPTIVE ONLINE CONTROL

1. By using the estimated submodels $\hat{f}(\cdot)$ and $\hat{g}(\cdot)$, compute the estimated system output as given in:

$$\hat{y}[n + d] = \hat{f}(x[n]) + \hat{g}(x[n])\mathbf{u}[\mathbf{n}] \tag{41}$$

– This is going to be used as the information on actual system output.

2. Compute the error $e_t[n]$. For trajectory tracking problem, the difference between the reference signal and the estimated system output is computed:

$$e_t[n] = r[n] - \hat{y}[n + d] \tag{42}$$

Compute the error $e_s[n]$. For stabilization problem the change in Lyapunov function is computed:

$$e_s[n] = \mathcal{V}(x_{n+1})^* - \mathcal{V}(x_n)^* \tag{43}$$

3. Update the weights (W^i) of the constructed recurrent neural network, described in detail above using problem specified error, in order to minimize the error:

$$W^i(n+1) = W^i(n) - \alpha P \sum_{k=1}^N \left| \frac{e_t}{N} \right| \tag{44}$$

or

$$W^i(n+1) = W^i(n) - \alpha P \sum_{k=1}^N \left| \frac{e_s}{N} \right| \tag{45}$$

4. The output of the recurrent neural network is P^* , the optimal P matrix. Use the computed weights to obtain the elements of P^* , matrix:

$$P^* = (W^i(n+1))^T P(W^i(n+1)) \tag{46}$$

5. Using the obtained P matrix, compute the inverse optimal control law:

$$u_n^* = \alpha(x_n) = -\frac{1}{2} (R(x_n) + P_2(x_n))^{-1} P_1(x_n) \tag{47}$$

– where

$$P_1(x_n) = \hat{g}^T(x_n) P^* \hat{f}(x_n) \tag{48}$$

and

$$P_2(x_n) = \hat{g}^T(x_n) P^* \hat{g}(x_n) \tag{49}$$

6. Apply this computed control input signal to the system to be controlled.

6 Simulation Results

The performance of the proposed neural network based conversion method from non-affine to affine system model using NARMA-L2 model and adaptive online inverse optimal control method has been tested by numerous simulations on two different benchmark systems. On each benchmark system, first the simulation results for the identification of the NARMA-L2 model by feedforward neural networks is depicted. The identification performance for

the benchmark systems can be observed in Figs. 5 and 12, respectively. In the identification step, an affine model is obtained for the non-affine system. Then the simulation results are illustrated for the adaptive inverse optimal control methodology where the optimal P matrix is evaluated online by recurrent neural networks. Additionally, simulations are repeated for the case when a sinusoidal disturbance with a magnitude of 2 units is applied and also for the case when a white noise with a signal-to-noise ratio (SNR) of 2 dB is applied to the system. It is assumed that disturbance and noise act on the system throughout the simulation. Consequently, the trajectory tracking problem is simulated for the nominal case when there is no disturbance or noise, for the case when disturbance is added to the system and also for the case when noise is added.

In a nutshell, the simulation results for the following cases are presented:

- (1) Offline system identification result.
- (2) Stabilization performance for the case with no disturbance or noise.
- (3) Stabilization performance when sinusoidal disturbance with a magnitude of 2 units is applied.
- (4) Stabilization performance when a white noise with an SNR of 2 dB is applied.
- (5) Trajectory tracking performance for the case with no disturbance or noise.
- (6) Trajectory tracking performance when sinusoidal disturbance with a magnitude of 2 units is applied.
- (7) Trajectory tracking performance when a white noise with an SNR of 2 dB is applied.

Besides giving the graphs of the simulation results, we also evaluated the performance of the proposed control method by computing a performance index. For this purpose, we calculated the integral square error (ISE) performance criterion for each benchmark system and for each case that was studied. The results are given in Tables 1 and 2. The integral square error (ISE) performance index is a function of the control error and it is formulated as:

$$\text{ISE} = \int e^2(t)dt \quad (50)$$

6.1 Benchmark System I

The first benchmark problem is a non-affine discrete-time system with delay where the input–output relation given by:

$$\begin{aligned} y(n+1) = & 0.2\cos(0.8y(n) + y(n-1)) + 0.4\sin\left(0.8(y(n) + y(n-1))\right) \\ & + 2u(n) + u(n-1) + 0.1(9 + y(n) + y(n-1)) \\ & + \frac{2(u(n) + u(n-1))}{10 + \cos(y(n))} \end{aligned} \quad (51)$$

Figure 5 shows the offline system identification result for benchmark system I. It is clearly observed from the figure that the proposed feedforward neural network structure can successfully identify the NARMA-L2 model of the system. The dynamics of this system is described by a single input–output relation, and the P matrix that is computed to assure stabilization is 1 by 1. Figure 6 illustrates the stabilization results for the nominal case. In Figs. 6, 7, 8, 9, 10 and 11, the system output is given together with the computed control input, the error and the adaptable P matrix component. Figure 7 depicts the performance results when a sinusoidal disturbance with a magnitude of 2 units is applied. Figure 8 depicts the stabilization results

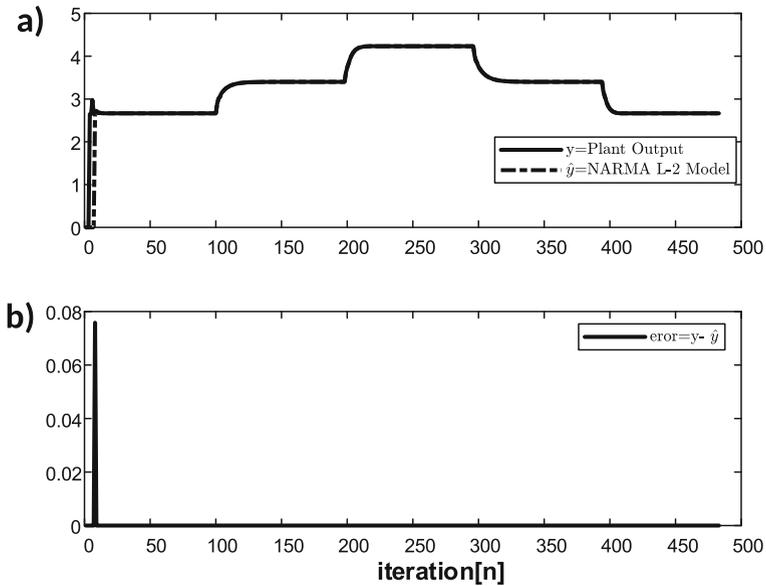


Fig. 5 System identification process based on NARMA-L2 modelling for benchmark system I, **a** System identification, **b** System identification error

Table 1 Results of simulations with respect to ISE performance index for benchmark system I given in (51)

Simulation\cases	Nominal	Measurement noise	Disturbance
Trajectory Tracking	2.36	3.83	2.42
Stabilization	4.850×10^{-8}	4.853×10^{-8}	4.858×10^{-8}

when a white noise with a 2 dB power is applied. The figures clearly show that the proposed control method can successfully provide good stabilization control and also can effectively deal with disturbance and noise. Figure 9 shows the performance of the proposed method for the trajectory tracking problem. It is clearly seen that the proposed control methodology can provide successful tracking and the transient effects are quickly diminished. Figure 10 illustrates the trajectory tracking performance of the system when a sinusoidal disturbance is applied and Fig. 11 depicts the case when noise is added.

In Table 1, the integral square error (ISE) performance index computed for each simulated case for benchmark system I is tabulated.

6.2 Benchmark System II

The second benchmark system is a second order non-affine plant that is characterized by the equations given in (52).

$$\begin{aligned}
 x_1(n + 1) &= x_2(n) \\
 x_2(n + 1) &= \sin(x_1(n)u(n)) + u(n)(5 + \cos(u(n)x_2(n))) \\
 y(n) &= x_1(n)
 \end{aligned}
 \tag{52}$$

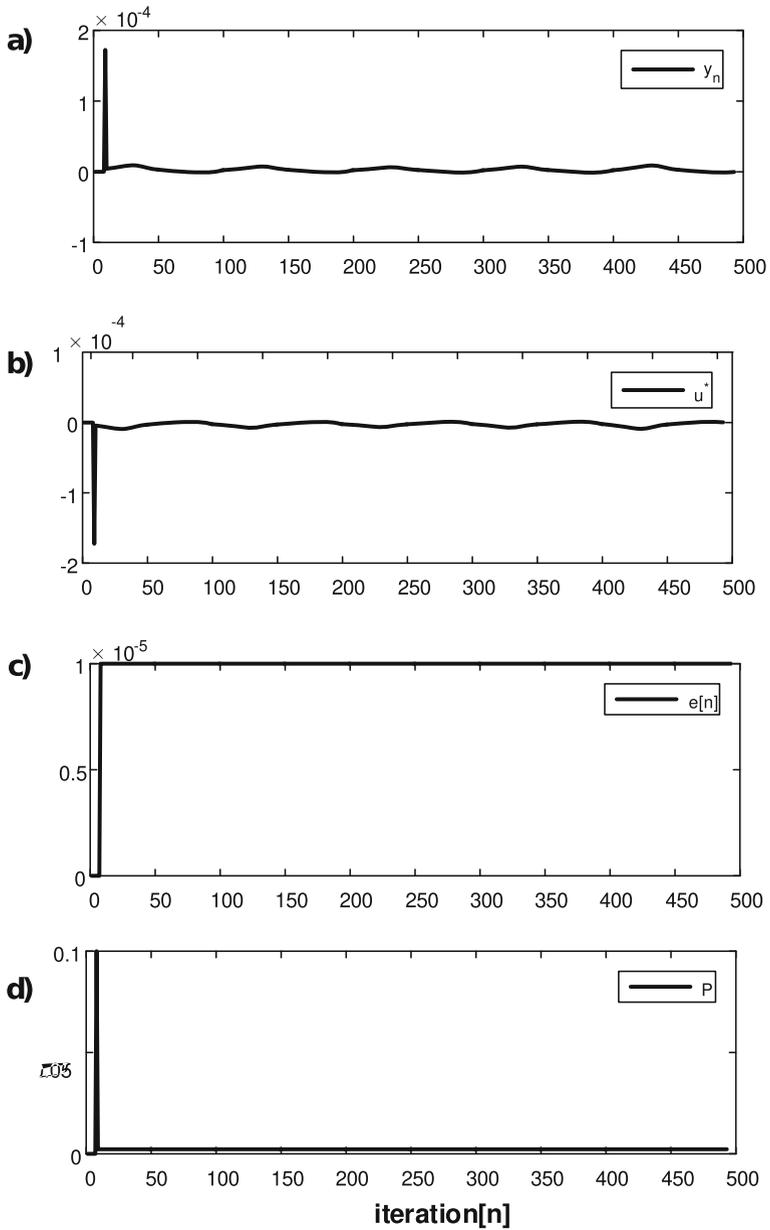


Fig. 6 Stabilization results for the nominal case for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** Error to the equilibrium point, **d** P parameter

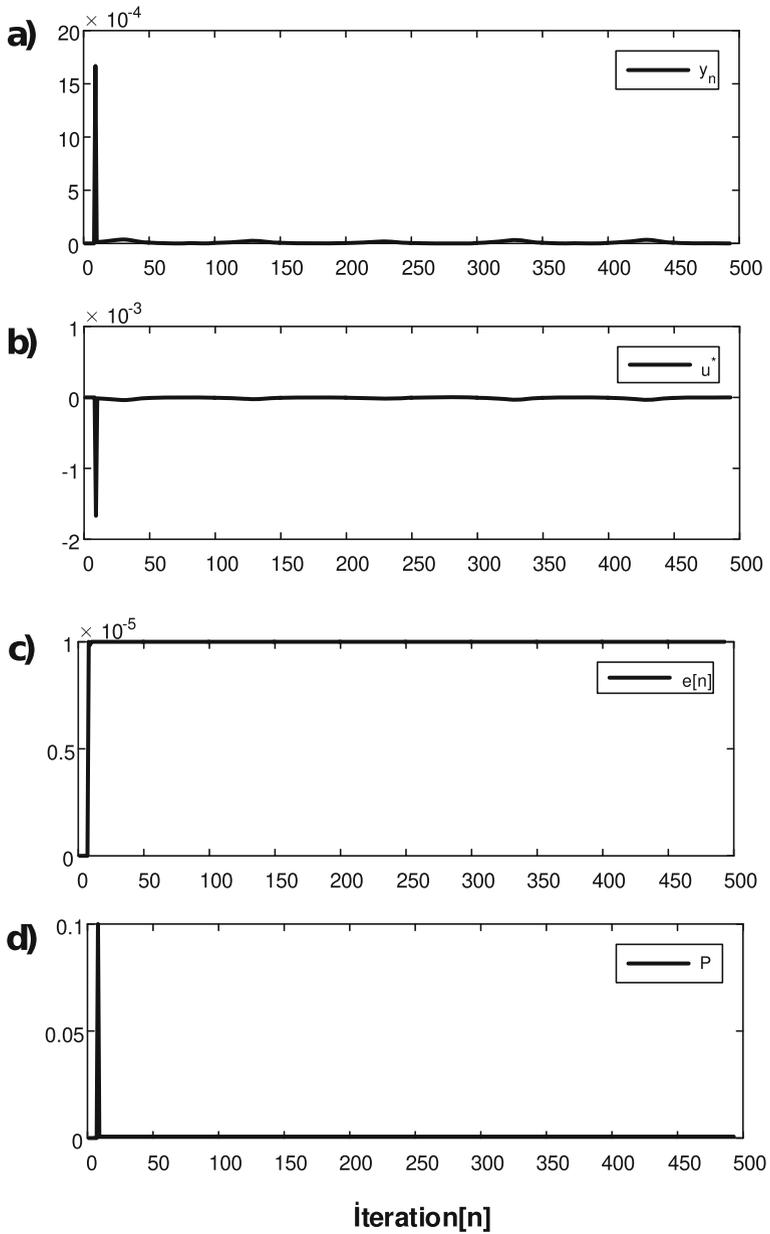


Fig. 7 Stabilization results when a sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** to the equilibrium point, **d** P parameter

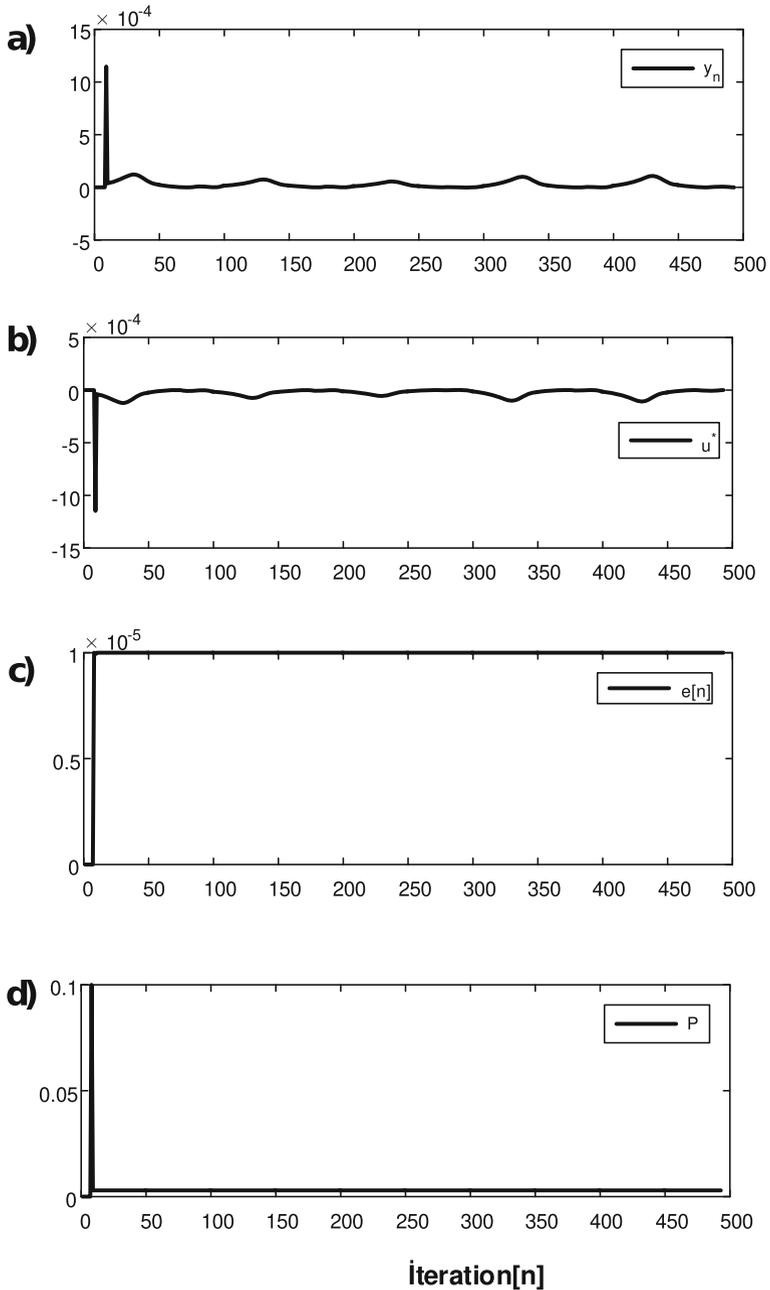


Fig. 8 Stabilization results with 2 dB (SNR) white noise for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** Error to the equilibrium point, **d** P parameter

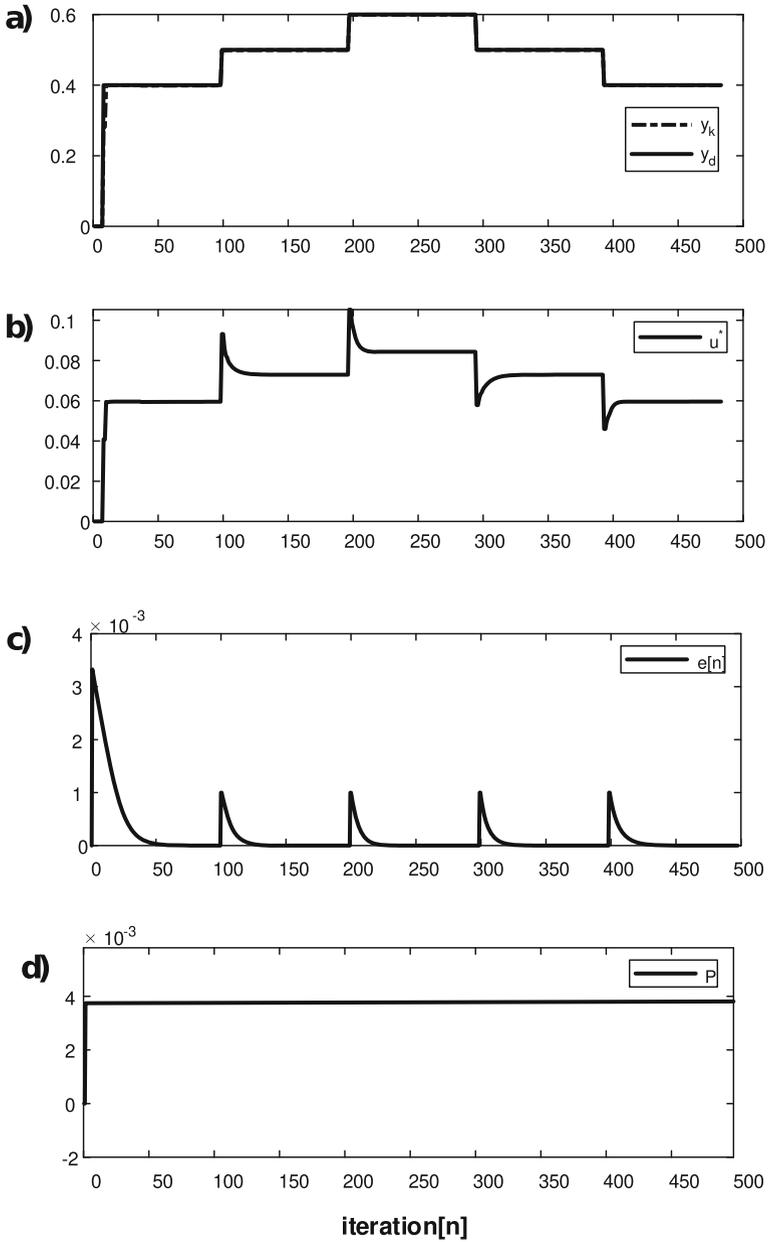


Fig. 9 Trajectory tracking results for the nominal case for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

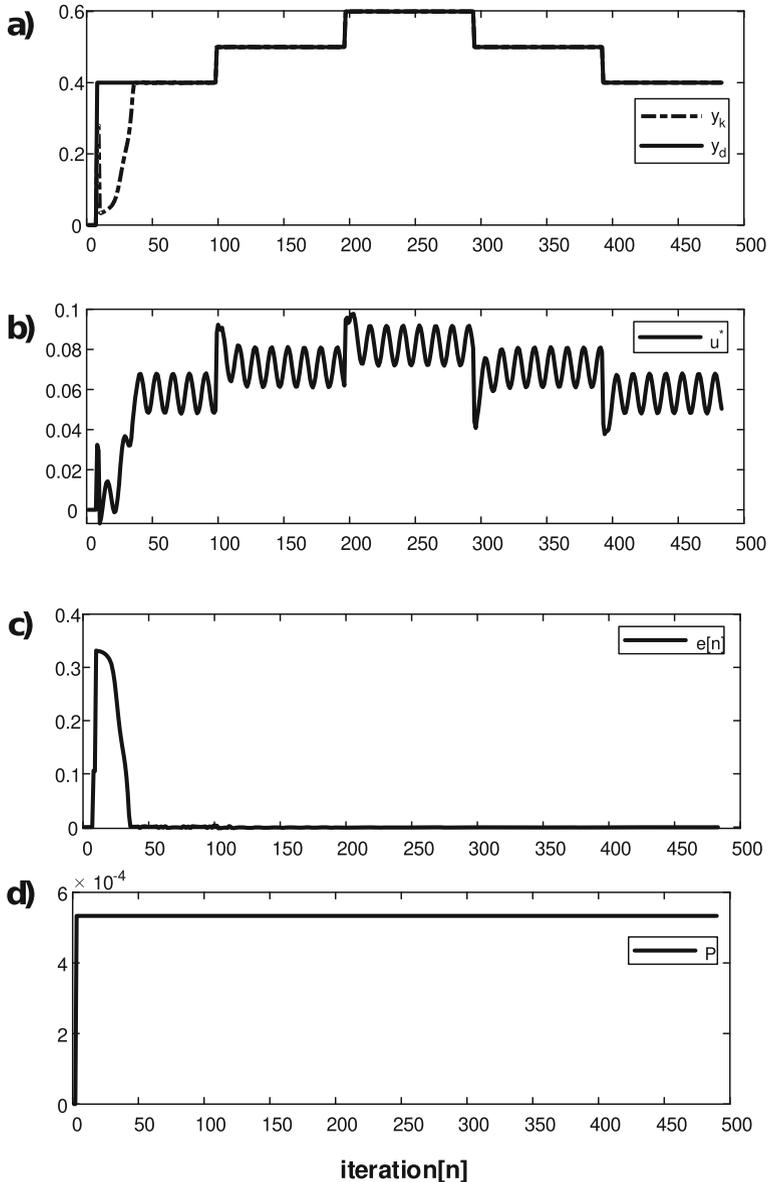


Fig. 10 Trajectory tracking results when sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

This system has two states, and the P matrix is a 2 by 2 matrix. To assure stabilization, the non-diagonal entries of the P matrix are assumed to be zero. Two separate neural networks have been designed to adaptively update the diagonal elements, P_{11} and P_{22} . Firstly, the offline system identification step has been carried out for benchmark system II and the result is illustrated in Fig. 12. The figure clearly shows the success of this identification process based on NARMA-L2 modelling. In Fig. 13, the stabilization results are given for the nominal case

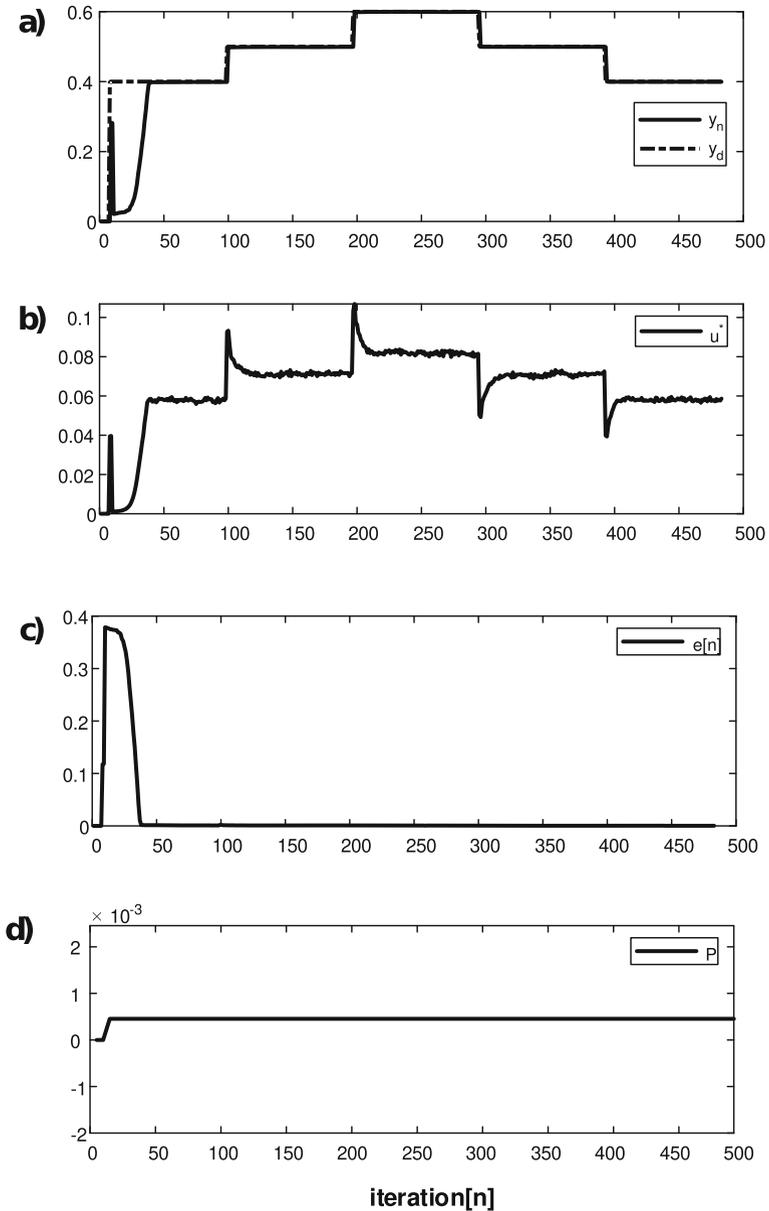


Fig. 11 Trajectory tracking results with 2 dB (SNR) white noise for benchmark system I. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

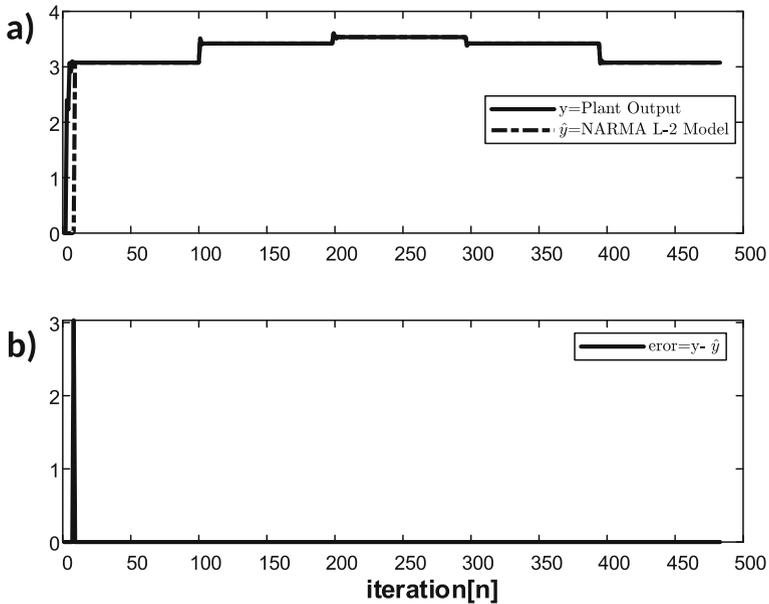


Fig. 12 System identification process based on NARMA-L2 modelling for benchmark system II, **a** System identification, **b** System identification error

where there is no disturbance or noise. In Figs. 13, 14 and 15, the results that are illustrated respectively are the states x_1 and x_2 , the control input u , the error in state x_1 (the error in state x_2 is not shown since it is one-step delayed version of the error for x_1), and the evolution of the diagonal terms of the P matrix, P_{11} and P_{22} . Figure 14 depicts the obtained results when a sinusoidal disturbance with a magnitude of 2 units is applied to the system and Fig. 15 shows the stabilization results when a white noise with a SNR of 2 dB is applied. It can be observed from these figures that the proposed control method is successful in stabilizing the system. Also, it is verified by the obtained results that the effects of the disturbance and noise can be successfully suppressed.

Next, benchmark system II is used to solve a trajectory tracking problem. State x_2 is selected as the system output to track a given reference trajectory. State x_1 is the one step delayed version of x_1 . In this case, the adaptable P matrix is 1 by 1, the recurrent neural network is used to optimize a single parameter. The trajectory tracking results for benchmark system II are illustrated in Figs. 16, 17 and 18. In these figures, the graphs of the system output, the computed inverse optimal control law, the trajectory tracking error and the P parameter are provided. In Fig. 16, the trajectory tracking results for the nominal case with no disturbance or noise are illustrated. In Fig. 17, results obtained for the case when a sinusoidal disturbance with a magnitude of 2 units is applied to the system are shown. Figure 18 depicts simulation results when a 2dB white noise is applied to the system. All the results justify that the proposed control method can successfully provide good trajectory control and also can effectively deal with the effects of disturbance and noise.

Additionally, the integral square error (ISE) performance index calculated for each simulated case for benchmark system II is given in Table 2.

Fig. 13 Stabilization results for the nominal case for benchmark system II, **a** State x_1 , **b** State x_2 , **c** Inverse optimal control law, **d** Error to the equilibrium point, **e** P_{11} element of P matrix, **f** P_{22} element of P matrix

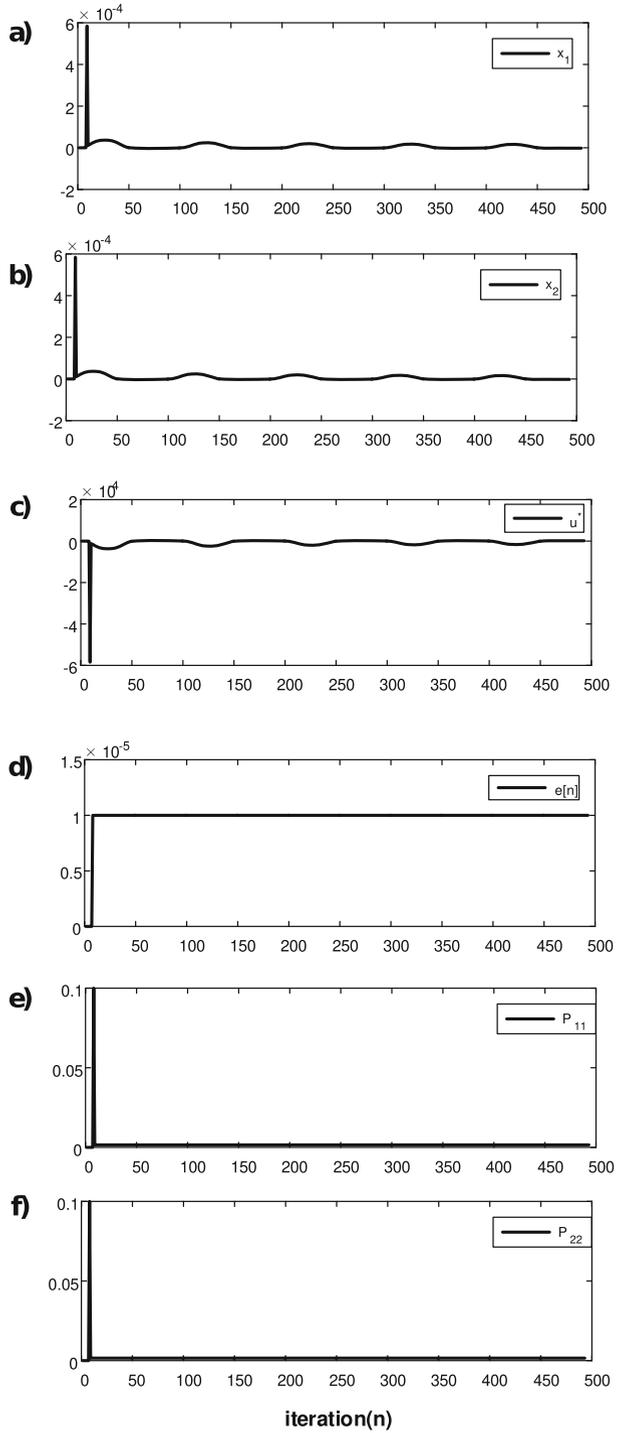
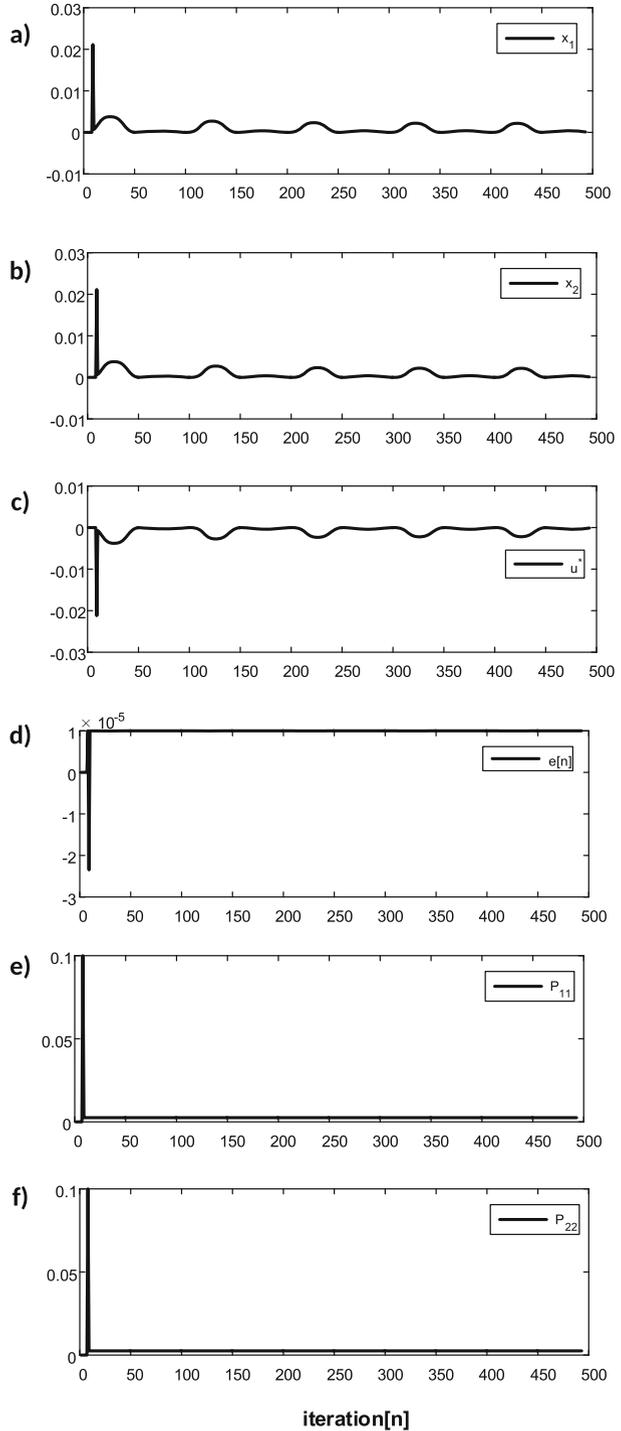


Fig. 14 Stabilization results when a sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system II. **a** State x_1 , **b** State x_2 , **c** Inverse optimal control law, **d** Error to the equilibrium point, **e** P_{11} element of P matrix, **f** P_{22} element of P matrix



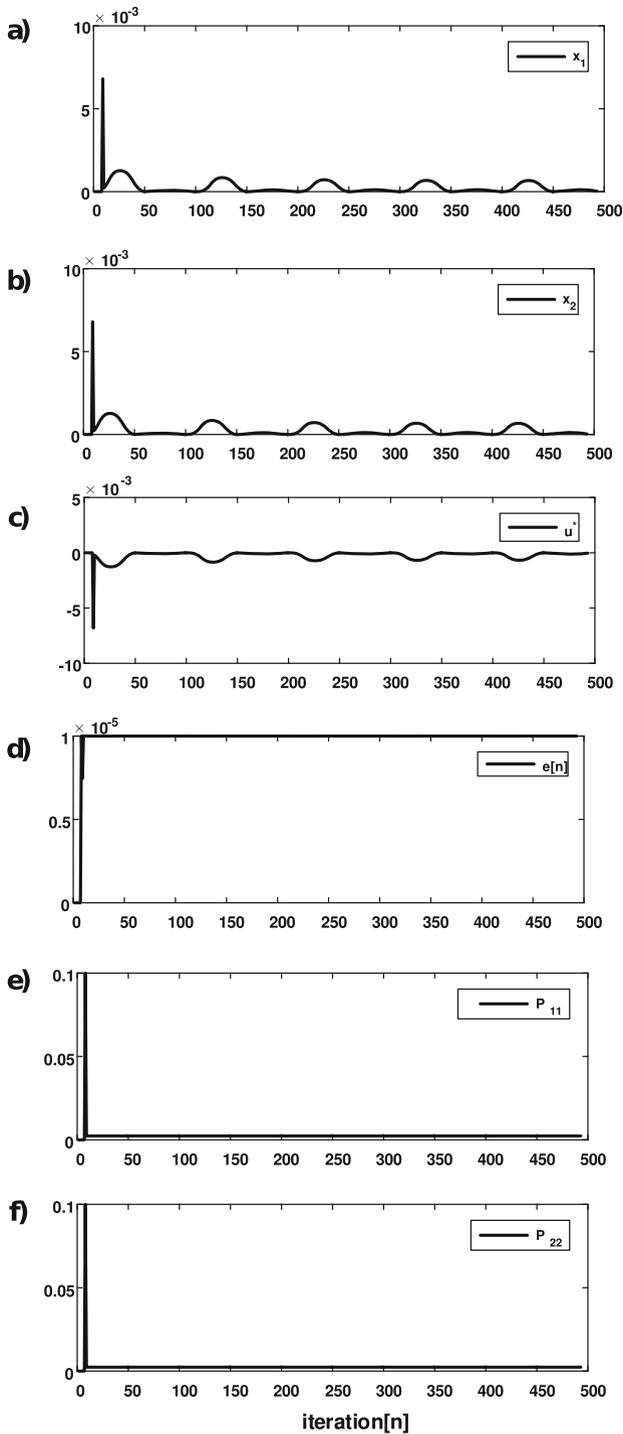


Fig. 15 Stabilization results with 2 dB (SNR) white noise for benchmark system II. **a** State x_1 , **b** State x_2 , **c** Inverse optimal control law, **d** Error to the equilibrium point, **e** P_{11} element of P matrix, **f** P_{22} element of P matrix

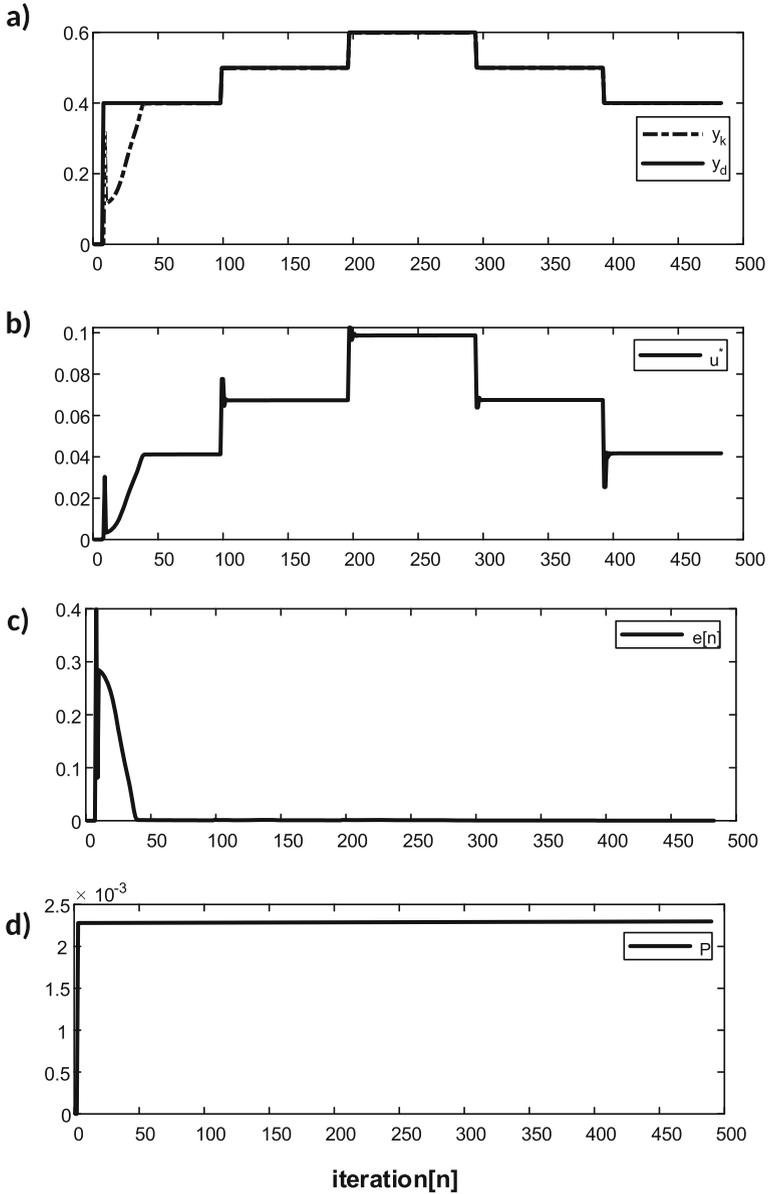


Fig. 16 Trajectory tracking results for the nominal case for benchmark system II. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

Table 2 Results of simulations with respect to ISE performance index for benchmark system II given in (52)

Simulation\cases	Nominal	Measurement noise	Disturbance
Trajectory tracking	1.46	3.12	2.35
Stabilization	4.855×10^{-8}	4.850×10^{-8}	4.898×10^{-8}

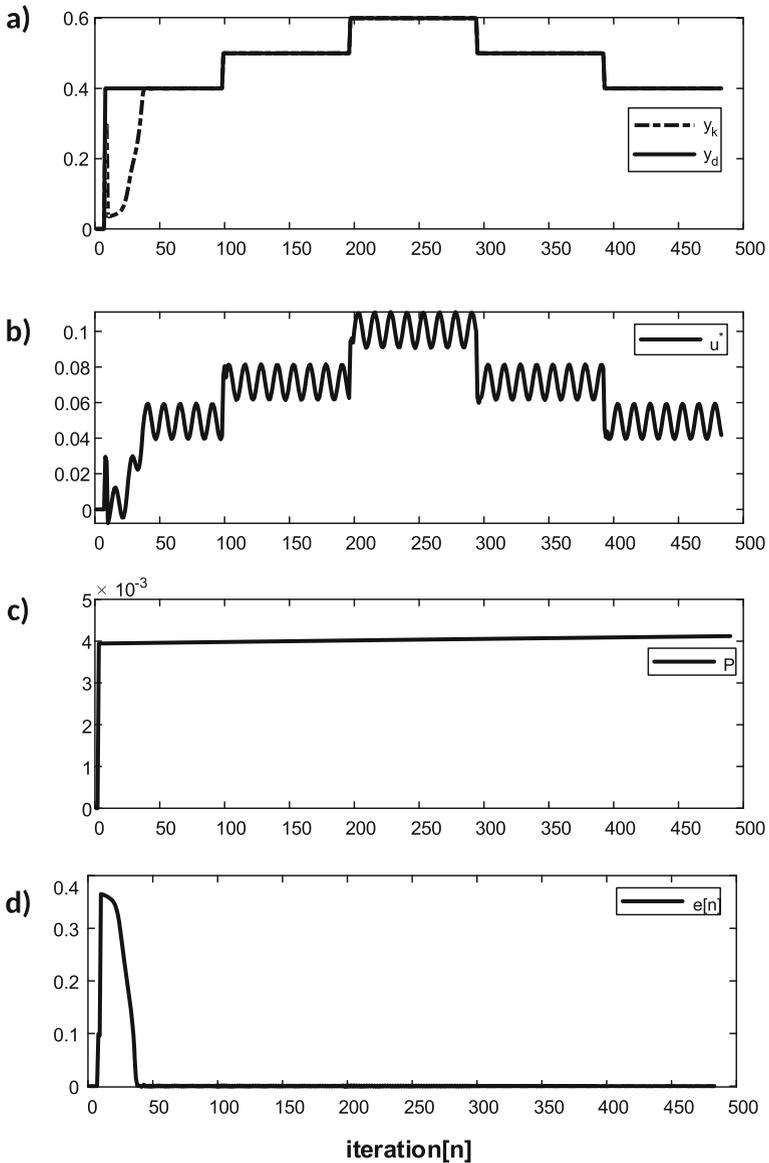


Fig. 17 Trajectory tracking results when a sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system II. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

7 Conclusion

In technical literature, there are numerous works on inverse optimal control of affine nonlinear systems, however there is no significant research on the application of inverse optimal control method on non-affine systems. In this paper, a novel control approach is proposed for non-affine nonlinear discrete time systems, based on inverse optimal control methodology.

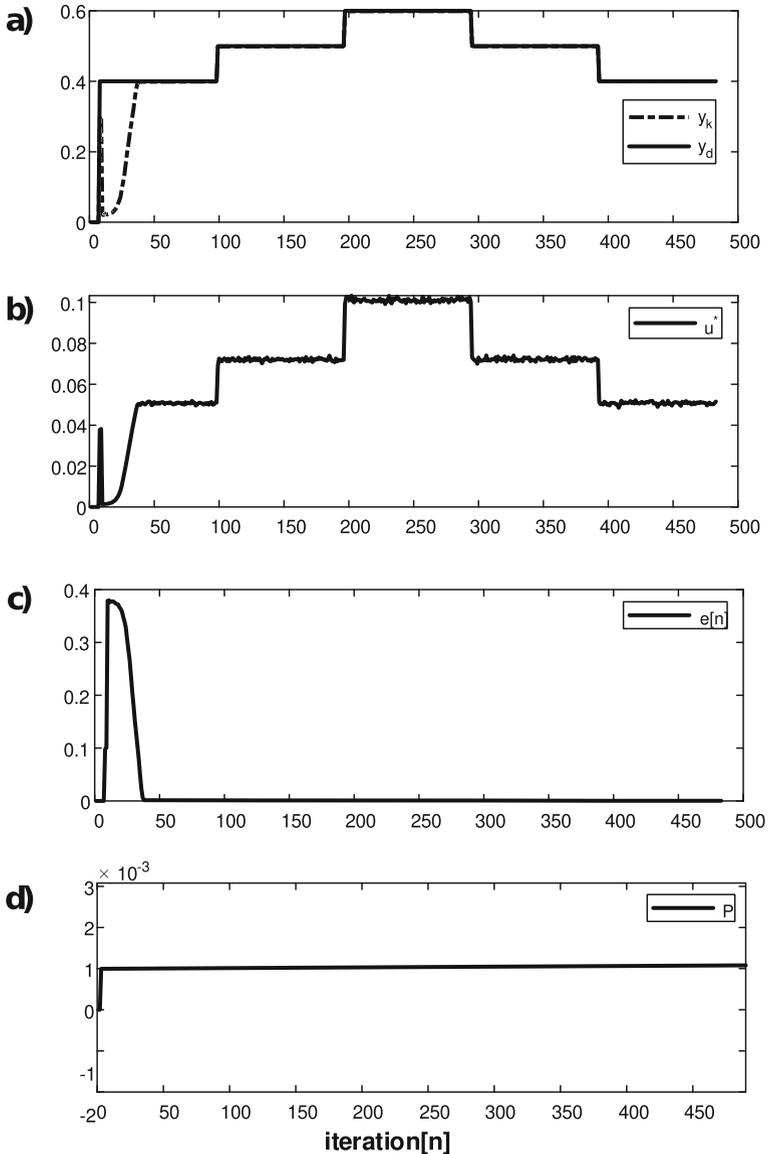


Fig. 18 Trajectory tracking results with 2 dB (SNR) white noise for benchmark system II. **a** System output, **b** Inverse optimal control law, **c** Tracking error, **d** P parameter

There are two major contributions of this work. The first one is that given the input–output data of a non-affine nonlinear system, its NARMA-L2 model is obtained using a multilayer feedforward controller. Hence the non-affine system is converted to an affine model. The modelling stage is carried offline. Next, the inverse optimal control law is obtained using the affine NARMA-L2 model of the system to be controlled by computing an adaptive and optimal P matrix using a neural network based approach. The design of the inverse optimal control law boils down to the selection of the optimal P matrix which is directly effective

on control performance. In this work an adaptive control architecture is proposed where the P matrix is computed online using a recurrent neural network, so P matrix is adapted at each time step in order to minimize a cost function derived from the control error. The extensive simulations performed on two different benchmark systems verify the success of the proposed control methodology.

Author Contributions SME wrote main manuscript text and all figures ,conducted research and simulations. GGO is advisor for the manuscript. All authors reviewed the manuscript.

Funding The author(s) received no financial support for the research, authorship, and/or publication of this article

Declarations

Conflict of interest The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ruiz-Cruz R, Sanchez EN, Ornelas-Tellez F, Loukianov AG, Harley RG (2013) Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator. *IEEE Trans Cybern* 43(6):1698–1709. <https://doi.org/10.1109/TSMCB.2012.2228188>
2. Ulusoy L, Güzelkaya M, Eksin İ (2019) Fusion of inverse optimal and model predictive control strategies. *Trans Inst Meas Control* 42:1122–1134. <https://doi.org/10.1177/0142331219884803>
3. Villaseñor C, Rios JD, Arana-Daniel N, Alanis AY, Lopez-Franco C, Hernandez-Vargas EA (2018) Germinal center optimization applied to neural inverse optimal control for an all-terrain tracked robot. *Appl Sci*. <https://doi.org/10.3390/app8010031>
4. Fotouhi R, Pourgholi M (2021) Discrete-time inverse optimal control for consensus of multi-agent systems via a novel meta-heuristic algorithm. In: 2021 7th International conference on control, instrumentation and automation (ICCIA), pp 1–5. <https://doi.org/10.1109/ICCIA52082.2021.9403537>
5. Perez-Villalpando M, Tun K, Muro C, Fausto F (2021) Inverse optimal control using metaheuristics of hydropower plant model via forecasting based on the feature engineering. *Energies* 14:7356. <https://doi.org/10.3390/en14217356>
6. Xue W, Kolaric P, Fan J, Lian B, Chai T, Lewis FL (2022) Inverse reinforcement learning in tracking control based on inverse optimal control. *IEEE Trans Cybern* 52(10):10570–10581. <https://doi.org/10.1109/TCYB.2021.3062856>
7. Neumeyer C, Oliehoek FA, Gavrila DM (2021) General-sum multi-agent continuous inverse optimal control. *IEEE Robotics Autom Lett* 6(2):3429–3436. <https://doi.org/10.1109/LRA.2021.3060411>
8. Li Y-M, Min X, Tong S (2020) Adaptive fuzzy inverse optimal control for uncertain strict-feedback nonlinear systems. *IEEE Trans Fuzzy Syst* 28(10):2363–2374. <https://doi.org/10.1109/TFUZZ.2019.2935693>
9. Li Y, Min X, Tong S (2021) Observer-based fuzzy adaptive inverse optimal output feedback control for uncertain nonlinear systems. *IEEE Trans Fuzzy Syst* 29(6):1484–1495. <https://doi.org/10.1109/TFUZZ.2020.2979389>
10. Min X, Li Y, Tong S (2020) Adaptive fuzzy output feedback inverse optimal control for vehicle active suspension systems. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2020.04.096>
11. Ricalde LJ, Sanchez E (2012) Inverse optimal neural control of a class of nonlinear systems with constrained inputs for trajectory tracking. *Optim Control Appl Methods*. <https://doi.org/10.1002/oca.986>

12. Denai MA, Palis F, Zeghib A (2004) Anfis based modelling and control of non-linear systems: a tutorial. In: 2004 IEEE International conference on systems, man and cybernetics (IEEE Cat. No.04CH37583), vol 4, pp 3433–34384. <https://doi.org/10.1109/ICSMC.2004.1400873>
13. Gretton A, Doucet A, Herbrich R, Rayner PJW, Scholkopf B (2001) Support vector regression for black-box system identification. In: Proceedings of the 11th IEEE signal processing workshop on statistical signal processing, pp 341–344. <https://doi.org/10.1109/SSP.2001.955292>
14. Rong H, Zhang G, Zhang C (2005) Application of support vector machines to nonlinear system identification. In: Proceedings autonomous decentralized systems, 2005. ISADS 2005, pp 501–507. <https://doi.org/10.1109/ISADS.2005.1452120>
15. Suykens JAK (2001) Nonlinear modelling and support vector machines. In: IMTC 2001. Proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics, vol 1, pp 287–2941. <https://doi.org/10.1109/IMTC.2001.928828>
16. Fotouhi R, Pourgholi M (2021) Discrete-time inverse optimal control for consensus of multi-agent systems via a novel meta-heuristic algorithm. In: 2021 7th International conference on control, instrumentation and automation (ICCIA), pp 1–5. <https://doi.org/10.1109/ICCIA52082.2021.9403537>
17. Atkinson C, Osseiran A (2011) Discrete-space time-fractional processes. *Fract Calc Appl Anal* 14:201–232. <https://doi.org/10.2478/s13540-011-0013-9>
18. Carrasco-Gutiérrez CE, Sosa W (2019) A discrete dynamical system and its applications. *Pesquisa Operacional* 39:457–469. <https://doi.org/10.1590/0101-7438.2019.039.03.0457>
19. Sanchez EN, Ornelas-Tellez F (2017) Discrete-time inverse optimal control for nonlinear systems. <https://doi.org/10.1201/b14779>
20. Galor O (2007) Discrete dynamical systems. Springer, Berlin. <https://doi.org/10.1007/3-540-36776-4>
21. Vega CJ, Suarez OJ, Sanchez EN, Chen G, Elvira-Ceja S, Rodriguez DI (2020) Trajectory tracking on uncertain complex networks via nn-based inverse optimal pinning control. *IEEE Trans Neural Netw Learn Syst* 31(3):854–864. <https://doi.org/10.1109/TNNLS.2019.2910504>
22. Rios J, Alanis A, Lopez-Franco M, Lopez-Franco C, Arana-Daniel N (2017) Real-time neural identification and inverse optimal control for a tracked robot. *Adv Mech Eng* 9:168781401769297. <https://doi.org/10.1177/1687814017692970>
23. Leon BS, Alanis AY, Sanchez EN, Ornelas-Tellez F, Ruiz-Velazquez E (2014) Neural inverse optimal control via passivity for subcutaneous blood glucose regulation in type 1 diabetes mellitus patients. *Intell Autom Soft Comput* 20:279–295. <https://doi.org/10.1080/10798587.2014.891307>
24. Lopez-Franco C, López-Franco M, Alanis A, Gómez-Avila J, Arana-Daniel N (2015) Real-time inverse optimal neural control for image based visual servoing with nonholonomic mobile robots. *Math Probl Eng* 2015:1–12. <https://doi.org/10.1155/2015/347410>
25. Mainprice J, Hayne R, Berenson D (2016) Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. *IEEE Trans Rob* 32(4):897–908. <https://doi.org/10.1109/TRO.2016.2581216>
26. Lopez V, Sanchez E, Alanis A, Rios J (2016) Real-time neural inverse optimal control for a linear induction motor. *Int J Control* 90:1–29. <https://doi.org/10.1080/00207179.2016.1213424>
27. Hernandez-Mejia G, Alanis A, Hernandez Vargas EA (2018) Neural inverse optimal control for discrete-time impulsive systems. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2018.06.034>
28. Gurubel K, Sanchez E, Coronado A, Zúñiga Grajeda V, Sulbaran B, Breton-Deval L (2019) Inverse optimal neural control via passivity approach for nonlinear anaerobic bioprocesses with biofuels production. *Optim Control Appl Methods*. <https://doi.org/10.1002/oca.2513>
29. Lopez-Garcia TB, Sanchez EN, Ruiz-Cruz R (2019) Real-time implementation of battery bank charge-discharge based on neural inverse optimal control. *IET Renew Power Gener* 13(16):3124–3132. <https://doi.org/10.1049/iet-rpg.2019.0581>
30. Lopez-Garcia TB, Sanchez EN, Ruiz-Cruz R (2019) Real-time implementation of battery bank charge-discharge based on neural inverse optimal control. *IET Renew Power Gener* 13(16):3124–3132. <https://doi.org/10.1049/iet-rpg.2019.0581>
31. Chan VM, Hernández-Vargas EA, Sánchez EN (2021) Neural inverse optimal control applied to design therapeutic options for patients with COVID-19. In: 2021 International joint conference on neural networks (IJCNN), pp 1–7. <https://doi.org/10.1109/IJCNN52387.2021.9534240>
32. Cai X, Lin C, Liu L, Zhan X (2018) Inverse optimal control for strict-feedforward nonlinear systems with input delays. *Int J Robust Nonlinear Control* 28(8):2976–2995. <https://doi.org/10.1002/rnc.4062>
33. Lin Z, Liu Z, Zhang Y, Chen CP (2022) Adaptive neural inverse optimal tracking control for uncertain multi-agent systems. *Inf Sci* 584:31–49
34. Ornelas F, Loukianov AG, Sanchez EN (2011) Discrete-time robust inverse optimal control for a class of nonlinear systems. *IFAC Proc Vol* 44(1):8595–8600. <https://doi.org/10.3182/20110828-6-IT-1002.03386>. 18th IFAC World Congress

35. Sanchez EN, Ornelas-Tellez F (2017) Discrete-time inverse optimal control for nonlinear systems. <https://doi.org/10.1201/b14779>
36. Leon BS, Alanis AY, Sanchez EN, Ornelas-Tellez F, Ruiz-Velazquez E (2012) Neural inverse optimal control applied to type 1 diabetes mellitus patients. <https://doi.org/10.1109/LASCAS.2012.6180310>. <https://ieeexplore.ieee.org/document/6180310>. Accessed 18 Oct 2022
37. Lewis FL, Syrmos VL (1995) Optimal control. A Wiley-interscience publication. Wiley, London
38. Kirk DE (2004) Optimal Control Theory: an Introduction. Courier Corporation
39. Sanchez EN (ed) (2018) Discrete-time recurrent neural control: analysis and applications. CRC Press Taylor and Francis Group, Boca Raton. <https://doi.org/10.1201/9781315147185>
40. Moulay E, Perruquetti W (2005) Stabilization of nonaffine systems: a constructive method for polynomial systems. *IEEE Trans Autom Control* 50:520–526. <https://doi.org/10.1109/tac.2005.844899>
41. Krstic M, Kanellakopoulos I, Kokotovic PV (1994) Nonlinear design of adaptive controllers for linear systems. *IEEE Trans Autom Control* 39(4):738–752. <https://doi.org/10.1109/9.286250>
42. Marino R, Tomei P (1996) Nonlinear control design: geometric, adaptive and robust. Prentice Hall International (UK) Ltd., GBR
43. Sastry SS, Isidori A (1989) Adaptive control of linearizable systems. *IEEE Trans Autom Control* 34:1123–1131. <https://doi.org/10.1109/9.40741>
44. Artstein Z (1983) Stabilization with relaxed controls. *Nonlinear Anal Theory Methods Appl* 7(11):1163–1173. [https://doi.org/10.1016/0362-546X\(83\)90049-4](https://doi.org/10.1016/0362-546X(83)90049-4)
45. Sontag ED (1983) A Lyapunov-like characterization of asymptotic controllability. *SIAM J Control Optim* 21:462–471. <https://doi.org/10.1137/0321028>
46. Uçak K, Günel G (2016) A novel adaptive narma-l2 controller based on online support vector regression for nonlinear systems. *Neural Process Lett* 44(3):857–886
47. Şen GD, Günel GÖ (2022) A NARMA-L2 controller based on online LSSVR for nonlinear systems. In: Zattoni E, Simani S, Conte G (eds) 15th European workshop on advanced control and diagnosis (ACD 2019). Springer, Cham, pp 213–231
48. Uçak K, Günel GÖ (2016) A novel adaptive NARMA-L2 controller based on online support vector regression for nonlinear systems. *Neural Process Lett*. <https://doi.org/10.1007/s11063-016-9500-7>
49. Celikel R, Aydogmus O (2018) NARMA-L2 controller for single link manipulator. In: 2018 International conference on artificial intelligence and data processing (IDAP), pp 1–6. <https://doi.org/10.1109/IDAP.2018.8620842>
50. Gundogdu A, Celikel R (2021) NARMA-L2 controller for stepper motor used in single link manipulator with low-speed-resonance damping. *Eng Sci Technol Int J* 24(2):360–371. <https://doi.org/10.1016/j.jestch.2020.09.008>
51. Narendra KS, Mukhopadhyay S (1997) Adaptive control using neural networks and approximate models. *IEEE Trans Neural Netw* 8(3):475–485. <https://doi.org/10.1109/72.572089>
52. Nørgård PM, Ravn O, Poulsen NK, Hansen L (2000) Neural networks for modelling and control of dynamic systems—a practitioner’s handbook
53. Ballini R, Von Zuben F (2000) Application of neural networks to adaptive control of nonlinear systems G. W. Ng, control systems centre series, UMIST, UK, 1997, ISBN: 0-86380-214-1. *Automatica* 36:1931–1933. [https://doi.org/10.5555/S0005-1098\(00\)00114-X](https://doi.org/10.5555/S0005-1098(00)00114-X)
54. Hagan MT, Demuth HB, Jesús OD (2002) An introduction to the use of neural networks in control systems. *Int J Robust Nonlinear Control* 12(11):959–985. <https://doi.org/10.1002/rnc.727>
55. Majstorovic M, Nikolic I, Radovic J, Kvascev G (2008) Neural network control approach for a two-tank system. In: 2008 9th Symposium on neural network applications in electrical engineering, pp 215–218. <https://doi.org/10.1109/NEUREL.2008.4685619>
56. Abu-Rub H, Awwad A (2009) Artificial neural networks and fuzzy logic based control of ac motors. In: 2009 IEEE International electric machines and drives conference, pp 1581–1586. <https://doi.org/10.1109/IEMDC.2009.5075414>
57. Almobaied M, Eksin I, Guzelkaya M (2019) Inverse optimal controller design based on multi-objective optimization criteria for discrete-time nonlinear systems. In: 2019 IEEE 7th Palestinian international conference on electrical and computer engineering (PICECE), pp 1–6. <https://doi.org/10.1109/PICECE.2019.8747189>
58. Almobaied M, Eksin I, Guzelkaya M (2018) Inverse optimal controller based on extended Kalman filter for discrete-time nonlinear systems. *Optim Control Appl Methods* 39(1):19–34. <https://doi.org/10.1002/oca.2331>
59. Akoum M, Günel GÖ (2021) Inverse optimal control based on improved grey wolf optimization algorithm. In: 2021 13th International conference on electrical and electronics engineering (ELECO), pp 47–51. <https://doi.org/10.23919/ELECO54474.2021.9677672>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.