ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ADAPTIVE INVERSE OPTIMAL CONTROLLER DESIGN FOR NON-AFFINE NONLINEAR SYSTEMS USING MACHINE LEARNING TECHNIQUES

Ph.D. THESIS

Muhammet Emre SANCI

Department of Mechatronics Engineering

Mechatronics Engineering Programme

MARCH 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ADAPTIVE INVERSE OPTIMAL CONTROLLER DESIGN FOR NON-AFFINE NONLINEAR SYSTEMS USING MACHINE LEARNING TECHNIQUES

Ph.D. THESIS

Muhammet Emre SANCI
(518162019)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Prof. Dr.Gülay ÖKE GÜNEL

MARCH 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANARAK DOĞRUSAL VE AFİN OLMAYAN SİSTEMLER İÇİN ADAPTİF TERS OPTİMAL KONTROLÖR TASARIMI

DOKTORA TEZİ

Muhammet Emre SANCI
(518162019)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr.Gülay ÖKE GÜNEL

MART 2024

Muhammet Emre SANCI, a PhD. student of ITU Graduate School student ID 518162019, successfully defended the thesis entitled "ADAPTIVE INVERSE OPTIMAL CONTROLLER DESIGN FOR NON-AFFINE NONLINEAR SYSTEMS USING MACHINE LEARNING TECHNIQUES", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr.Gülay ÖKE GÜNEL** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Müjde GÜZELKAYA** Istanbul Technical University	.....
	Asst.Prof. Figen ÖZEN Halic University	.....
	Asst.Prof. İlker ÜSTOĞLU Istanbul Technical University	.....
	Prof. Dr. Cenk ULU Yildiz Technical University	.....

Date of Submission : 12 February 2024
Date of Defense : 22 March 2024

FOREWORD

The successful completion of this PHD thesis is attributed to the continuous support and the outstanding research environment provided by various individuals.
In particular, I want to convey my deepest and sincere appreciation to my academic supervisor, Prof. Dr. Gulay ÖKE GÜNEL, for her enthusiasm, enlightening guidance, patience, constructive comments, and unwavering support throughout my doctoral program.
I also express my gratitude for the assistance provided by Assoc. Prof. Kemal UÇAK; each meeting with him added invaluable aspects to the implementation and broadened my perspective.

MARCH 2024

Muhammet Emre SANCI
(Research Assistant)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANFIS :Adaptive Neuro Fuzzy Inference Systems
ANN :Artificial Neural Network
ARE :Algebraic Riccati Equation
BP :Backpropagation
CLF :Control Lyapunov Function
CSTR :Continuously Stirred Tank Reactor
DT :Discrete-Time
ESS :Error sum-of-squares
HJB :Hamilton-Jacobi-Bellman Equation
IOC :Inverse Optimal Control
LM :Levenberg-Marquadt
LQR :Linear Quadratic Regulator
LSSVR :Least Squares Support Vector Regression
NARMA :Nonlinear Autoregressive Moving Average
NARX :Nonlinear Autoregressive Network With Exogenous Inputs
PID :Proportional-Integral-Derivative
RE :Riccati Equation
RHONN : Recurrent High Order Neural Network
SARS-CoV-2 :Severe Acute Respiratory Syndrome Corona Virus 2
SISO :Single-Input, Single-Output
SVR :Support Vector Regressor
T1DM :Type 1 Diabetes Mellitus

SYMBOLS

$\mathbf{C}$	:Optimal Value Function
$\mathbf{z_n}$	:Tracking Error
$\mathbf{u_{c_n}}$	:Inverse Optimal Control Law With Integrator
$\mathbf{x}_{\delta,n}$	:Desired Trajectory
$\mathbf{H}$	:Hamiltonian
$\mathbf{d}$	:Relative Degree
$\hat{\mathbf{f}}, \hat{\mathbf{g}}$	: NARMA-L2 model's Submodels
N	:Size Of Training Data
$\mathbf{L}$	:Lagrangian
$\mathbf{J_m}$	:Jacobian
$\mathbf{x_n}$	:System State
$\mathbf{u_n}$	:Control Input
$\mathbf{y_n}$	:System Output
$\alpha_{\,\mathrm{k}}$	:Lagrange Multipliers
$\mathbf{n_u}$	:Number Of Past Inputs
$\mathbf{n_y}$	:Number Of Past Plant Outputs
$\mathbf{K_I}$	:Integrator Gain
$\mathbf{K_P}$	:Propptoional Gain
$\mathbf{K_D}$	:Derivative Gain
$\mathbf{V}$	:Lyapunov Function
$\mathbf{T}$	:Sampling Interval
$\mathbf{n}$	:Time Index
$\mathbf{k}$	:Number Of Training Data
$\hat{\mathbf{y}}_{\mathbf{NARMA_n}}$	:Narma Model Output
$\hat{\mathbf{y}}_{\mathbf{NARX}}$	:Narx Model Output
$\mathbf{d_{n+k,j}}$	:Euclidean Distance Between The State Vector And jth support vector
$\delta\mathbf{u_n}$	:Correction Term
$\mathbf{P}$	:Inverse Optimal Controller Weighting Paramter
$\mathbf{R}$	:Inverse Optimal Controller Gain
$\mathbf{f,g}$	:System Dynamics
$\mathbf{K}$	: Estimation Horizon
λ	:Penalty Term
$\mathbb{R}^n$	:Real n-Dimensional Vector Space
$\mathbb{R}^m$	:Real m-Dimensional Vector Space
$\mathbb{R}^{n\times m}$	:Real $n \times m$ Dimensional Vector Space
σ	:Kernel Parameter
$\mathbf{w}$	:Weight Vector
$\varphi(\mathbf{x_k})$	:Nonlinear Mapping Function
$\mathbf{e_k}$	:Error Of Least-Squares Data Fitting
$\Omega_\mathbf{n}$	:Kernel Matrix
μ_1, μ_2	:Regressor Parameters

LIST OF TABLES

Page

LIST OF FIGURES

Page

ADAPTIVE INVERSE OPTIMAL CONTROLLER DESIGN FOR NON-AFFINE NONLINEAR SYSTEMS USING MACHINE LEARNING TECHNIQUES

SUMMARY

The primary aim of the optimal control problem is to derive a control input function or control law that meets specific control and state constraints while optimizing a given performance criterion. However, obtaining the optimal control rule is a challenging task, even in unconstrained and linear cases, and often cannot be achieved analytically. The solution typically involves solving the Hamilton-Jacobi-Bellman (HJB) equation, a complex task, especially when dealing with nonlinear systems where an analytical HJB solution is often unavailable. In linear systems with a quadratic performance criterion, the HJB equation transforms into a Riccati equation, presenting its own challenges for analytical solutions. An alternative perspective to tackling optimal control problems is through the theory of Inverse Optimal Control (IOC), which circumvents the need to solve the intricate HJB equation. The inverse optimal control problem perspective is a concept introduced by Kalman in the early 1960s. Kalman defined it as follows: Given a dynamic system and a feedback control law, if the closed-loop system is asymptotically stable, the inverse problem is to search for the most general performance index for which this control law is optimal. In fact, IOC is an approach rather than a methodology that perceives the optimal control problem in reverse. In the IOC method, control Lyapunov function (CLF) based control approaches are widely used when the controller is desired to be stable and optimal according to meaningful objective functions. The formulation of CLF, which provides the design of the most suitable feedback controller for commonly known typical system classes, has been discussed extensively in the literature. Its existence implies the stability of the system. Therefore, the distinctive aspect of this approach is the subsequent determination of the performance measure corresponding to the feedback control that stabilizes the system. Since there is no clear and precise technique for determining the CLF for general nonlinear systems, the most challenging aspect of IOC is the determination of the CLF itself. IOC has been increasingly used in recent years to solve nonlinear optimal control problems in many real-time applications. IOC provides an alternative solution to optimal control problems in nonlinear systems by saving the trouble of solving the HJB equation. However, another point that should be noted is that there is currently not enough work on the optimal control of nonlinear systems. In studies using IOC, systems with affine inputs are preferred. It is not possible to solve the Hamilton-Jacobi-Bellman (HJB) equations, especially for non-affine nonlinear systems. There has been no research on the use of IOC on non-affine nonlinear systems.

IOC has been used in many systems by avoiding the HJB problem, which is cumbersome to solve, but due to IOC structure, its use is limited to systems whose

input and output relationship is linear and whose input signal can be isolated from the system dynamics (non-affine non-linear systems). However, in reality, the input and output relationships of many systems are both non-linear and not isolable from the system dynamics. In most studies in the literature, it is generally preferred to use non-linear affine input systems in control theory and related fields due to various reasons such as analytical simplicity, ease of control design, computational efficiency, and modeling approach. However, while systems with nonlinear affine inputs offer these advantages, it is important to recognize that not all systems can be accurately represented by models with nonlinear affine inputs. In cases where systems exhibit strong nonlinear properties, it may be necessary to use more complex and accurate nonlinear system models with non-affine inputs, although this poses additional analytical and computational challenges. In order to increase the applicability of IOC and expand its application area, and because there is no research in this topic in the literature, this thesis focuses on the solution of the problem of using IOC on non-affine controlled nonlinear systems. In the thesis, first of all, the basic concepts constituting the background material that has been utilized is explained. Then, a comprehensive explanation on IOC has been propvided. Consequently, two different methods were proposed in the thesis to use IOC on non-affine nonlinear systems.

In this context, this thesis describes a new inverse optimal controller design using the nonlinear autoregressive moving average-L2 (NARMA-L2) modeling technique and the offline artificial neural networks (ANN) method as the first method. First, the nonlinear autoregressive external input (NARX) model of the system is obtained using the offline ANN method, and a mapping of the system inputs and outputs is generated. Then, this model is decomposed into NARMA-L2 submodels, again using ANN. As a result, the non-affine system model converges to an affine system model. These resulting NARMA-L2 submodels then play an important role in calculating the inverse optimal control rule. In addition, the parameter of the inverse optimal controller is tuned online using recurrent neural networks. The performance of the ANN-based NARMA-L2 model and the proposed inverse optimal controller was evaluated by simulations and comparisons performed on two reference systems using MATLAB. The results of these simulations show that the ANN-based NARMA-L2 model and the inverse optimal controller achieve highly successful modeling and control performance.

The second proposed method describes another new inverse optimal controller utilizing the NARMA-L2 modeling technique and the online least squares support vector regression (LSSVR) method. The process begins with deriving the nonlinear autoregressive with exogenous inputs (NARX) model using the online LSSVR method. The model is then disintegrated into NARMA-L2 submodels, transforming the non-affine system model into a nonlinear affine system model. These NARMA-L2 submodels play a crucial role in calculating the inverse optimal control law. The parameters of the inverse optimal controller are fine-tuned online using the Levenberg-Marquardt algorithm. To the best of our knowledge, the method proposed in this thesis is one of the three studies in the literature that adjusts all parameters of IOC online, and it is the only study so far using it on non-affine nonlinear systems. The performance of the proposed inverse optimal controller based on

the NARMA-L2 model obtained by LSSVR and the inverse optimal controller was evaluated by simulations and comparisons performed on two reference systems in the MATLAB environment. Simulation and comparison studies for proposed method on two benchmark systems demonstrate that the proposed inverse optimal controllers exhibit commendable modeling and control performances.

In both studies, simulation experiments conducted on MATLAB have confirmed the effectiveness and performance of the proposed methods. As a possible future extension of this thesis, the aim is to explore the use of different machine learning-based algorithms and to develop new adaptive inverse optimal controller structures that do not require NARMA-L2 transformation.

MAKİNE ÖĞRENMESİ TEKNİKLERİ KULLANARAK DOĞRUSAL VE AFİN OLMAYAN SİSTEMLER İÇİN ADAPTİF TERS OPTİMAL KONTROLÖR TASARIMI

ÖZET

Optimal kontrol probleminin temel amacı, belirli kontrol ve durum kısıtlamalarını karşılayan ve bir performans kriterini optimize eden bir kontrol giriş fonksiyonu veya kontrol kuralı elde etmektir. Ancak, kısıtsız ve doğrusal durumlarda bile, optimal kontrol kuralını analitik olarak elde etmek basit değildir. Optimal kontrol kuralını bulmak genellikle Hamilton-Jacobi-Bellman (HJB) denklemini çözmeyi gerektirir ki bu da oldukça karmaşık bir süreçtir. Özellikle doğrusal olmayan sistemlerde, genellikle analitik bir HJB çözümü bulunmamaktadır. Sistem doğrusal olduğunda ve performans kriteri ikinci dereceden olduğunda, HJB denklemi bir Riccati denklemine dönüşür ve bu da belirli durumlarda analitik çözümlerle uğraşmayı zorlaştırır. Öte yandan, zorlayıcı HJB problemi yerine farklı bir bakış açısı sunan bir başka yaklaşım ise Ters Optimal Kontrol (TOK) teorisidir. Ters optimal kontrol problemi perspektifi, Kalman'ın 1960'ların başlarında ortaya koyduğu bir kavramdır. Kalman'ın tanımına göre, bir dinamik sistem ve bir geri besleme kontrol yasası verildiğinde ve kapalı çevrim sistemi asimptotik olarak kararlı olduğunda, ters problem, bu kontrol yasasının optimal olduğu en genel performans indeksini aramaktır. Aslında TOK, optimal kontrol problemini tam tersinden algılayan bir metodolojiden ziyade bir yaklaşım olarak görülebilir. TOK yönteminde kontrolörün kararlı ve anlamlı amaç fonksiyonlarına göre optimal olması istendiğinde kontrol Lyapunov fonksiyonu (KLF) tabanlı kontrol yaklaşımları yaygın olarak kullanılmaktadır. Yaygın olarak bilinen tipik sistem sınıfları için en uygun geri beslemeli kontrolörün tasarımını sağlayan KLF'nin formülasyonu literatürde çokca ele alınmıştır. KLF'nin varlığı, sistemin kararlılığını gösterir. Bu nedenle, bu yaklaşımın ayırt edici yönü, sistemi kararlı kılan geri besleme kontrolüne karşılık gelen performans ölçütünün sonradan belirlenmesidir. Genel doğrusal olmayan sistemler için KLF'nin belirlenmesine yönelik açık ve kesin bir teknik bulunmadığından, TOK'un en zorlu yönü KLF'nin kendisinin belirlenmesidir. TOK, son yıllarda birçok gerçek zamanlı uygulamada doğrusal olmayan optimal kontrol problemlerini çözmek için giderek daha fazla kullanılmıştır. TOK, HJB denklemini çözme zahmetinden kurtararak, doğrusal olmayan sistemlerde optimal kontrol problemlerine alternatif bir çözüm sağlar. Ancak belirtilmesi gereken bir diğer nokta ise şu anda doğrusal olmayan ve afin olmayan sistemlerin optimal kontrolü için yeterli çalışmanın bulunmamasıdır. Afin sistemler dinamik denklemlerde kontrol girdisinin lineer olarak ayrıştırılabildiği sistemlerdir. Eğer kontrol girdisi lineer olarak ifade edilemiyorsa sisteme afin olmaya sistem denir. TOK kullanılarak yapılan çalışmalarda afin girişli sistemler tercih edilmiştir. Özellikle afin olmayan kontrollü doğrusal olmayan sistemlere ilişkin Hamilton-Jacobi-Bellman (HJB) denklemlerinin

çözülmesi mümkün değildir. TOK'un afin olmayan kontrollü doğrusal olmayan sistemler üzerinde kullanımına ilişkin bir araştırma yapılmamıştır.

TOK çözülmesi külfetli olan HJB probleminden kaçınmayı sağlayarak bir çok sistemde kullanılmıştır fakat yapısı itibariyle kullanım alanı sistem modeli olarak giriş ve çıkış ilişkisi doğrusal ve giriş işareti sistem dinamiklerinden izole edilebilen sistemler ile sınırlı kalmıştır. Ancak gerçekte pek çok sistemin giriş ve çıkış ilişkisi hem doğrusal değildir hem de sistem dinamiklerinden izole edilebilir değildir (afin olmayan ve doğrusal olmayan sistemler). Literatürde yapılan çalışmalar esnasında analitik basitlik, kontrol tasarımı kolaylığı, hesaplama verimliliği, modelleme yaklaşımı gibi çeşitli nedenlerden dolayı genellikle kontrol teorisi ve ilgili alanlarda doğrusal olmayan afin girişli sistemlerin kullanılması tercih edilmektedir. Ancak doğrusal olmayan afin girişli sistemler bu avantajları sunarken, tüm sistemlerin doğrusal olmayan afin girişli modellerle doğru şekilde temsil edilemeyeceğini kabul etmek önemlidir. Sistemlerin güçlü doğrusal olmayan özellikler sergilediği durumlarda, ilave analitik ve hesaplama zorlukları oluşturmasına rağmen, daha karmaşık ve doğru, afin olmayan doğrusal olmayan sistem modellerinin kullanılması gerekli olabilir. Hem TOK'un uygulanabilirliğinin artırılması ve kullanım alanının genişletilmesi açısından hem de literatürde bu yönde bir araştırma yapılmamış olması açısından, bu tezde TOK'un afin olmayan ve doğrusal olmayan sistemler üzerinde kullanılabilmesi probleminin çözümü üzerine yoğunlaşılmıştır. Tezde öncelikle çalışmada kullanılmış olan temel kavramlar anlatılmıştır. Daha sonra TOK üzerine kapsamlı bir anlatım yapılmıştır. Ardından tezde TOK'un afin olmayan kontrollü doğrusal olmayan sistemler üzerinde kullanılabilmesi üzerine iki farklı yöntem önerilmiştir.

Bu bağlamda, ilk yöntem olarak nonlineer otoregresif hareketli ortalama-L2 (NOHO-L2) modelleme tekniği ve çevrimdışı yapay sinir ağları (YSA) yöntemi kullanılarak tasarlanmış yeni bir ters optimal kontrolör tasarımı anlatılmıştır. İlk olarak, sistemin doğrusal olmayan otoregresif dışsal girişli (DOODG) modeli çevrimdışı YSA yöntemi kullanılarak elde edilir, sistem giriş ve çıkışlarının haritalandırması oluşturulur. Ardından, yine YSA kullanılarak bu model NOHO-L2 alt modellerine ayrılır. Sonuç olarak, afin olmayan sistem modeli, afin bir sistem modeline yakınsar. Daha sonra bu elde edilen NOHO-L2 alt modelleri, ters optimal kontrol kuralının hesaplanmasında önemli bir rol oynar. Buna ek olarak, ters optimal kontrolörün parametresi yinelemeli sinir ağları kullanılarak çevrimiçi olarak ayarlanır. YSA tabanlı NOHO-L2 modeli ve optimal kontrole dayalı önerilen ters optimal kontrolörün performansı, MATLAB kullanılarak iki referans sistem üzerinde gerçekleştirilen simülasyonlar ve karşılaştırmalarla değerlendirilmiştir. Bu simülasyonların sonuçları, YSA tabanlı NOHO-L2 modeli ve ters optimal kontrolorün, yüksek başarılı bir modelleme ve kontrol performansı elde ettiğini göstermektedir.

Önerilen ikinci yöntem hareketli ortalama-L2 (NOHO-L2) modelleme tekniği ve çevrimiçi en küçük kareler destek vektör regresyonu (EKKDVR) yöntemi kullanılarak tasarlanmış başka bir ters optimal kontrolör yapısıdır. Öcelikle, sistemin doğrusal olmayan otoregresif dışsal girişli (DOODG) modeli çevrimiçi EKKDVR yöntemi kullanılarak elde edilir. Ardından, model NOHO-L2 alt modellere ayrılır. Sonuç olarak, afin olmayan sistem modeli, afin bir sistem modeline yakınsar. Daha sonra bu

elde edilen NOHO-L2 alt modelleri, ters optimal kontrol kuralının hesaplanmasında kullanılır. Ayrıca, ters optimal denetleyicinin parametreleri Levenberg-Marquardt algoritması kullanılarak çevrimiçi olarak ayarlanır. Bildiğimiz kadarıyla önerdiğimiz yöntem literatürde TOK'ün bütün parametrelerini çevrim içi ayarlayan üç çalışmadan biri, ayrıca afin olmayan kontrollü doğrusal olmayan sistemler üzerinde TOK kullanılarak yapılan şu ana kadarki tek çalışmadır. EKKDVR tabanlı NOHO-L2 modeli ve ters optimal denetleyiciye dayalı önerilen ters optimal kontrolörün performansı, MATLAB ortamında iki referans sistem üzerinde gerçekleştirilen simülasyonlar ve karşılaştırmalarla değerlendirilmiştir. Bu simülasyonların sonuçları, EKKDVR tabanlı NOHO-L2 modeli ve ters optimal denetleyicinin, çok daha yüksek başarılı bir modelleme ve kontrol performansı elde ettiğini göstermektedir.

Her iki çalışmada da MATLAB üzerinde yapılan benzetimler önerilen yöntemlerin işlerliğini ve başarımını teyit etmiştir. Bu tez üzerine olası gelecek geliştirme çalışması olarak farklı makine öğrenimi tabanlı algoritmaların kullanımı ve NOHO-L2 dönüşümü gerektirmeyen yeni uyarlanabilir ters optimal kontrolör yapılarının oluşturulması önerilebilir.

1. INRODUCTION

The optimal control problem arises from defining an objective function within the context of control system dynamics, along with a set of constraints or specifications. For many centuries, academics have been interested in the topic of optimal control. A major and frequently used component of optimal control is the calculus of variations since the 17th century. Later in the 19th century, Jacobi, Hamilton, and Weierstrass improved it further, after the work of Euler and Lagrange in the 18th century. Significant progress in mathematical aspects of optimal control was achieved in the 20th century [3–6]. First, it was proposed and explored how to optimize linear-stationary control systems under quadratic performance indices and bounded control effort constraints [7]. The suggested solution was then developed by Rekasius and Hsia through establishing essential and sufficient conditions for the existence of the saturation type optimal control rules [8]. In a sub-chapter of the calculus of variations, the inverse problems were studied with a focus on the forms to find Lagrangians given a family of curves. The inverse problem was first introduced in the context of dynamic programming and automatic control to find the criterion function given an optimal policy and the descriptive equations. The inverse problems were studied by emphasizing the forms to determine Lagrangians given a family of curves. [9]. Under the assumptions of linear plant and control law, measurable state variables, quadratic loss functions with constant coefficients, and a single control variable, Kalman developed the first formulation of the inverse optimal control. He did this by formulating, studying, and solving the inverse problem of optimal control theory in an effort to find all performance indices given a control law [10].

Ornelas et al. [11] introduced a discrete-time inverse optimal control (IOC) problem centered around achieving output tracking for a nonlinear system. The proposed method did not tackle a burdensome HJB solution. It required the system's

sub-dynamics to solve optimal control problems. However, it was limited to only systems represented in an affine form where control input and system output show linear relation. In reality, many systems do not show linear input and output behavior, and dynamics are intermingled. In this thesis, we focused on the problem of applying the proposed inverse optimal control by Ornelas at al. to non-affine systems in which system input and output show non-linear relation behavior and intermingled with system dynamics.

Inverse optimal control relies on the information of the system model to compute the control input, therefore it can be regarded as a model-based control method. The accuracy of model estimation has a major impact on performance in model-based adaptive control techniques. The system model must be obtained with high precision to achieve accurate control. Machine learning-based modeling techniques are frequently used to achieve this objective. Model-based control approaches that rely on models have successfully used artificial neural networks (ANN) [12–15], support vector regressors (SVR) [16–18], and adaptive neuro-fuzzy inference systems (ANFIS) [14, 15, 19] for model identification.

In general, machine learning-driven system identification methods start by postulating a model structure and subsequently attempt to pinpoint the model's input-output relationship. This group of predictive model structures includes the so-called NARMA-L2 model. Owing to its practical attributes, the NARMA-L2 model has lately been used in diverse applications. NARMA-L2 modeling involves expressing the nonlinear model through a Taylor series expansion while preserving the first-order terms. A common practice in the literature involves deriving NARMA-L2 models for systems through the application of an offline training stage [2, 20–27]. Considering the control system design, the calculation of the control input holds vital importance. In affine systems, the control input can be distinguished from the system dynamics and manifests in a linear fashion, simplifying the computation of the control input.

Conversely, in models of non-affine systems, the control input becomes intertwined with the system dynamics, preventing their distinct separation. Not only is the study

of non-affine discrete-time systems fascinating conceptually but also it presents an intricate and demanding area of study. Devising control input methods for non-affine systems presents a formidable challenge. While the adaptive control literature extensively addresses discussions and analyses concerning diverse nonlinear systems and methodologies, it is important not to overlook research pertaining to non-affine systems, as they hold greater universality when compared to their affine counterparts [28]. A large portion of physical systems possess intrinsic non-affine characteristics. Currently, diverse control approaches have been formulated to devise different control strategies to address nonlinear non-affine systems, such as backstepping control [29–32], adaptive control [33–35], optimal control [36–42], fuzzy control [43–46], neural control [47–51], etc.

The inverse optimal control method finds application in both nonlinear deterministic systems and nonlinear stochastic systems, encompassing scenarios involving additive noises, parametric uncertainties, and disturbances. The inverse optimal control (IOC) technique utilizes model information to calculate the control law, as an example of a model-based control method. Direct utilization of optimal control theory leads to the derivation of the Hamiltonian-Jacobi-Bellman (HJB) equation. When dealing with linear systems, solving the HJB equation leads to a straightforward solution known as the linear quadratic regulator (LQR) problem. Despite the favorable performance of the LQR technique, the conventional process of choosing the weight matrices (Q and R) in the LQR controller still relies on a trial-and-error approach guided by the designer's experience. Consequently, the ongoing research focus lies in optimizing these matrices, as they directly influence control performance [52]. Dealing with nonlinear problems makes the task of solving the HJB equation quite challenging. As a result, the approach of inverse optimal control has been developed to sidestep the requirement of deriving the precise solution for the HJB equation in the context of nonlinear systems.

The literature contains numerous research investigations that have been carried out utilizing the inverse optimal control approach [53–58]. Numerical techniques and global optimization methodologies have been combined with inverse optimal control

to find the ideal parameter values [59–62]. The integration of neural networks [63–72] along with other approaches rooted in machine learning [73–76] have been utilized together with the inverse optimal control methodology.

The inverse optimal control method can be applied to affine nonlinear systems, where the control input can be separated from the nonlinear dynamics, as shown by the papers focused in the technical literature cited above. Nevertheless, there needs to be more research on using inverse optimal controllers to control non-affine nonlinear systems.

This thesis presents two novel approaches for utilizing the inverse optimal control method in the control of nonlinear and non-affine systems. The first method utilizes neural networks in model identification and computation of the control law. Initially, the inherent non-affine system model is transformed into an affine model using the NARMA-L2 modeling technique by employing a feedforward neural network, the NARMA-L2 model facilitates the conversion to an affine-like representation. Consequently, this adaptation enables the utilization of the inverse optimal control formulation specifically derived for affine nonlinear systems. Subsequently, a control Lyapunov function is developed by employing a recurrent neural network. This online neural network continuously computes the Lyapunov function. As a result, the proposed methodology integrates the inverse optimal control approach into an adaptive control architecture. The adaptive nature of the system lies in the continuous utilization of the neural network to facilitate the implementation of the inverse optimal control method.

The second approach proposed in this thesis is based on utilizing LSSVR method for obtaining the NARMA-L2 model and designing the control law. Initially the NARX model corresponding to the target system is derived. Subsequently, this NARX model is deconstructed into a NARMA-L2 model. Both the NARX and NARMA-L2 models are identified using the online LSSVR method, which is fine-tuned using the Levenberg-Marquardt algorithm. The NARMA-L2 decomposition facilitates the transition from a non-affine system model to an affine one. Following this, based on the established affine system model, the inverse optimal controller is

developed. The controller and model parameters have been fine-tuned utilizing the Levenberg-Marquardt method. The outcome control framework is an adaptive structure in which the system model and control rule are derived at each sampling time in an online adaptive way. Because the success of model-based control techniques relies on the accuracy of the model, dynamically adjusting the system model at each time step enhances the method's likelihood of achieving success. At last, an integrator controller has been incorporated into the second control architecture to reduce the observed steady-state error. SVR-based approaches offer a significant advantage over identification algorithms based on backpropagation by ensuring global extremum, which enables the precise identification of the system model across all regions. Consequently, this thesis presents an innovative online inverse optimal controller for single-input, single-output (SISO) nonlinear and non-affine dynamical systems.

The proposed research in this thesis makes several noteworthy improvements to the technical literature. These contributions can be outlined as follows:

1) Two brand-new approaches are introduced for applying the inverse optimal control methodology to nonlinear and non-affine systems. As far as our understanding goes, this thesis represents the initial implementations of the inverse optimal control technique in non-affine systems.

2) An inverse optimal control method based on neural networks has been introduced for non-affine nonlinear systems. This method involves utilizing the NARMA-L2 modeling technique to transform the initial non-affine system model into an affine system model. Specifically, a feedforward neural network is employed to acquire the NARMA-L2 model. Additionally, a recurrent neural network has been utilized for the calculation of inverse optimal controller's parameter.

3) For the second proposed approach, an online least squares support vector regression (LSSVR) has been integrated with the inverse optimal control method. Online LSSVR serves to transform the initial non-affine system model (NARX) into an affine system model by employing the NARMA-L2 modeling approach.

4) Another notable breakthrough in this thesis is the adaptivity introduced to both the P and R parameters of the inverse optimal controller. These parameters

are adaptively updated in an online fashion through iterative processes using the Levenberg-Marquardt algorithm.

5) Through the utilization of the inverse optimal control methodology alongside the online LSSVR technique, an adaptive control structure is established. Within this framework, model identification and control design are conducted iteratively online.

Moreover, as far as our understanding extends, this thesis represents the inaugural incorporation of the inverse optimal control methodology into the literature in an adaptive mode, including the optimization of parameters at every sampling time for nonlinear non-affine systems.

This thesis is structured as follows: Chapter 2, briefly describes useful and necessary mathematical concepts, introduces a concise knowledge of nonlinear and non-affine systems, and details the principles of NARMA-L2 modeling. Chapter 3, explains the inverse optimal control method for trajectory tracking. Chapter 4, gives a brief explanation about backpropagation neural networks and a detailed description of the first proposed method, simulation results and performance analysis are represented. In Chapter 5, online LSSVR is outlined and a detailed explanation of the second proposed control architecture is presented, simulation results and performance analysis of the method are provided. Simulations of the suggested approach have been evaluated through two separate benchmark problems for each of the methods. The outcomes of the simulations validate that the introduced control methods are capable of effectively achieving precise tracking control. Moreover, the robustness of the method is also confirmed by the simulation results obtained under disturbance, noise, and parametric uncertainty cases. Furthermore, an extensive and thorough comparison is carried out between the proposed second control strategy and a traditional PID controller in order to compare the performance.

2. MATHEMATICAL PRELIMINARIES

This chapter provides a brief overview of key concepts utilized in this thesis such as optimal control theory, Lyapunov stability, passivity, basics of non-affine systems and NARMA-L2 modeling.

2.1 Optimal Control

Consider the following nonlinear, discrete, and affine system [77–79]:

$$x_{n+1} = f(x_n) + g(x_n)u_n \tag{2.1}$$

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

The cost function pertaining to the system's trajectory and control input is structured as outlined below:

$$C(z_n) = \sum_{i=n}^{\infty} (l(z_i) + u_i^T R u_i) \tag{2.2}$$

where $z_n = x_n - x_{\delta,n}$ represents the tracking error across the trajectory x_n, and $x_{\delta,n}$ represents the desired trajectory of x_n. Here, superscripts denote dimensions. $R : \mathbb{R}^n \to \mathbb{R}^{m\times m}$ expresses a weighting matrix which is real, symmetric and positive definite; $C(z_n) : \mathbb{R}^n \to \mathbb{R}^+$, $l(z_n) : \mathbb{R}^n \to \mathbb{R}^+$ is a positive semi-definite function. It is possible to set the components of the R matrix as constants or they can be dependent on the state of the system so that the weighting on the control rule can be varied based on the state value. Within the state feedback control basis, it is assumed that all possible values of state x_n can be accessed. The cost function $C(z_n)$ in equation (2.2) can be used as a Lyapunov function and it can be rewritten as follows:

$$\begin{aligned} V(z_n) &= (l(z_n) + u_n^T R u_n) + \sum_{i=n+1}^{\infty} (l(z_i) + u_i^T R u_i) \\ &= (l(z_n) + u_n^T R u_n) + V(z_{n+1}) \end{aligned} \tag{2.3}$$

In order to classify V as a Lyapunov function, the boundary condition $V(0) = 0$ must hold. The function $V(z_n)$ becomes time-invariant in the context of infinite horizon optimization and satisfies the discrete-time (DT) Bellman equation that ensures Bellman optimality principle [80].

$$V(z_n) = \min_{u_n}[(l(z_n) + u_n^T R u_n) + V(z_{n+1})] \tag{2.4}$$

where $V(z_{n+1})$ depends on $x_n, x_{\delta,n}$ and u_n by taking one step ahead prediction for z_{n+1}. The DT Bellman equation is resolved backward in time, which is an important aspect to take into account. The discrete-time Hamiltonian $H(z_n, u_n)$ is constructed as follows to achieve the requirements that the optimal control law must satisfy [81]:

$$H(z_n, u_n) = (l(z_n) + u_n^T R u_n) + V(z_{n+1}) - V(z_n) \tag{2.5}$$

Since $\frac{\partial H(u_n, z_n)}{\partial u_n} = 0$ is a requirement for the control law to be optimal, this indicates:

$$0 = 2Ru_n + \frac{\partial V(z_{n+1})}{\partial u_n} \tag{2.6}$$

In (2.6), the chain rule for vectors is employed to obtain:

$$0 = 2Ru_n + g^T(x)\frac{\partial V(z_{n+1})}{\partial z_{n+1}} \tag{2.7}$$

The HJB equation is obtained by applying optimal control theory to the trajectory tracking problem as follows:

$$l(z_n) + V(z_{n+1}) - V(z_n) + \frac{1}{4}\frac{\partial V^T(z_{n+1})}{\partial z_{n+1}} g^T(x_n) R^{-1}(z_n) g(x_n) \frac{\partial V(z_{n+1})}{\partial z_{n+1}} = 0 \tag{2.8}$$

Solving (2.8) is not a straightforward or easy task. This constraint represents a central hurdle in the optimal control of nonlinear discrete-time systems. To overcome this hurdle, the inverse optimal control approach has been proposed in the literature.

2.2 Lyapunov Stability

It is useful to introduce the definitions below as the inverse optimal control relies on the implementation of a Lyapunov function.

Definition 2.2.1: Radially Unbounded Function A positive definite function $V(x_n)$ satisfying $V(x_n) \to \infty$ as $\|x_n\| \to \infty$ is referred to as radially unbounded.

Definition 2.2.2: Decrescent Function A function $V : \mathbb{R}^{\times} \to \mathbb{R}$ is considered to be decrescent if there exists a positive definite function β such that the following inequality holds:

$$V(x_n) \leq \beta(\|(x_n)\|), \forall n \geq 0. \tag{2.9}$$

Theorem 2.1: Global Asymptotic Stability The equilibrium point $x_n = 0$ of (2.1) is globally asymptotically stable if there exists a function $V : \mathbb{R}^n \to \mathbb{R}$ such that
(i) V is a positive definite, decrescent and radially unbounded function.
(ii) $-\Delta V(x_n, u_n)$ is a positive definite function, where $\Delta V(x_n, u_n) = V(x_{n+1}) - V(x_n)$.

Definition 2.2.3: Control Lyapunov Function Assuming that $V(x_n)$ meets the radially unbounded function property, let's say that $V(0) = 0$, $\forall x_n \neq 0$, and $V(x_n) > 0$.If for any $x_n \in \mathbb{R}^n$ real values u_n exist such that $\Delta V(x_n, u_n) < 0$, where the Lyapunov difference 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 as a discrete-time control Lyapunov function (CLF) for (2.1).

2.3 Passivity

Assume the following nonlinear, discrete, affine system [77–79]:

$$x_{n+1} = f(x_n) + g(x_n)u_n \tag{2.10}$$

$$y_n = h(x_n) + J(x_n)u_n \tag{2.11}$$

Here, $x_n \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$, $u_n \in \mathbb{R}^m, \to \mathbb{R}^{n \times m}$, $g : \mathbb{R}^n$, $J : \mathbb{R}^n \to \mathbb{R}^{m \times m}$ $y_n \in \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^m$. In this definition, the state is x_n, the control input is u_n, and the system

output is y_n. $f(.)$, $h(.), g(.)$, and $J(.)$ are symmetric functions with $f(0) = 0$, $h(0) = 0$, $n \in Z^+$ and for all $x_n \neq 0$, $g(x_n) \neq 0$.

Definition 2.3.1 Passivity: If there exists a nonnegative function $V(x_n)$, as the storage function, such that for all u_n

$$V(x_{n+1}) - V(x_n) \leq y_n^T u_n \tag{2.12}$$

then the system described by equations (2.10)-(2.11) is called passive.

Definition 2.3.2: Feedback Passive System If a passivation law

$$u_n = \alpha(x_n) + v_n, \alpha, V \in \mathbb{R}^m, \tag{2.13}$$

is present with a smooth function of mapping $\alpha(x_n)$ and a function of storage $V(x_n)$, such that the system (2.10) with (2.13), expressed by

$$x_{n+1} = \bar{f}(x_n) + g(x_n)v_n, x_0 = x(0) \tag{2.14}$$

and output

$$\bar{y_n} = \bar{h}(x_n) + J(x_n)v_n, \tag{2.15}$$

establishes (2.12) with v_k as the new input, in which $\bar{f}(x_n) = f(x_n) + g(x_n)\alpha(x_n)$ and $\bar{h} : \mathbb{R}^n \to \mathbb{R}^m$ as a smooth function of mapping, with $\bar{h}(0) = 0$, then

the system described by (2.10)-(2.11) is feedback passive.

An essential point to consider is that the output responsible for rendering the system passive might not necessarily be the variable we intend to control. Instead, it is employed solely for the purpose of control synthesis. In essence, the process of making system (2.10) feedback passive can be described as identifying a suitable passivation law, u_n, and an output, $\bar{y}_n$, in a way that ensures the fulfillment of relation (2.12) according to the new input, v_n.

2.4 Nonlinear Non-Affine Systems

A significant class of nonlinear systems is non-affine systems. Non-affine systems demonstrate nonlinearity in relation to the control input, whereas affine systems are characterized by their linearity in terms of control input, which is present in a linear

form in the equations modeling these systems.

A nonlinear and affine continuous system is widely represented as follows:

$$\dot{x} = f(x) + g(x)u \tag{2.16}$$

$x \in \mathbb{R}^n$ represents the vector of system states, and $u \in \mathbb{R}^m$ represents the vector of control input signal. By separating the control input from the nonlinear dynamics, this representation eventually leads to a linear form for the control input. As a result, the design of the control law is simplified. Due to the inherent nonlinearity of most physical systems, it is not possible to accurately model them under the assumption of being affine. Systems that exhibit the characteristic of the control input being inseparable from the internal dynamics are known as non-affine systems. The given expression presents the most comprehensive formulation for non-affine nonlinear systems:

$$\dot{x} = f(x,u) \tag{2.17}$$

Equation (2.17) may be rebuilt in Taylor expansion form as follows when $f(x,u)$ is a smooth function: (2.18) [82]

$$\dot{x} = f_0(x) + \sum_{j=1}^{k} f_j(x)u^{[j]} + R(x,u) \tag{2.18}$$

Therefore, it is possible to approximate nonlinear systems by utilizing affine system models via including only the first-order terms mentioned in equation (2.18).

Currently, there is lack of sufficient research on optimal control for non-affine nonlinear systems. The Riccati Equation (RE), Algebraic Riccati Equation (ARE), and Hamilton-Jacobi-Bellman (HJB) equations for control systems are difficult to solve because non-linear non-affine systems present more difficulties than linear or nonlinear affine control systems [83]. The structure of the plant model, which deviates from the standard affine system format due to the nonlinear appearance of the control input u, presents the primary obstacle in solving the control problem of nonlinear non-affine systems. Due to the complex nature of non-affine systems, there has been limited research on Lyapunov-based controllers for these systems. Instead,

the majority of controllers designed for non-affine systems rely on non-model-based techniques, such as fuzzy logic or neural network controllers [84–86]. In certain papers, authors have suggested various transformations to convert non-affine systems into an affine structure. These approaches are prone to uncertainties in state-space equations, potentially resulting in suboptimal performance and even system instability [87]. Due to that, it is also a common approach to make the assumption that the plant to be controlled is affine, implying that the model exhibits linearity with respect to the control input. However, it should be noted that most real physical systems are inherently nonlinear. In such systems, it is not possible to express the control input separately from the nonlinear dynamics [82, 88–92]. While the development of nonlinear analytical controllers for non-affine systems holds significant value, there is a limited body of literature dedicated to addressing nonlinear Lyapunov-based control methods for these systems [93–96]

As a consequence, it can be concluded that designing control laws for discrete-time non-linear non-affine systems is a challenging task due to the presence of nonlinearity in either the control input or the system output. This research introduces a new adaptive inverse optimal control approach specifically designed for non-affine systems. The methodology involves deriving the NARMA-L2 model of the system under control, effectively transforming the original nonlinear non-affine system model into an affine system model and applying a Lyapunov-based control method.

2.5 NARMA-L2 System Identification

NARMA is an abbreviation for the nonlinear autoregressive moving average, which offers a discrete-time model of a nonlinear dynamical system within the proximity of its equilibrium state [97].

A general expression for a non-affine and nonlinear discrete-time system can be written as:

$$y_{n+d} = F_{actual}(u_n, \cdots, u_{n-k+1}, y_n, \cdots, y_{n-k+1}) \tag{2.19}$$

Note that here, y_n stands for the system output, u_n represents the control input and d shows the relative degree. The Taylor series expansion of (2.19) is written down,

keeping the first order terms, to generate the NARMA-L2 model. Thus, the following is the derivation of the NARMA-L2 model:

$$\begin{aligned}\hat{y}_{n+d} &= \hat{f}[y_n, y_{n-1}, \cdots, y_{n-k+1}, u_{n-1}, \cdots, u_{n-k+1}] \\ &+\hat{g}[y_n, y_{n-1}, \cdots, y_{n-k+1}, u_{n-1}, \ldots, u_{n-k+1}] \cdots u_n\end{aligned} \tag{2.20}$$

where

$$\begin{aligned}\hat{f} &= F_{actual}[(y_n, \cdots, y_{n-k+1}, 0, u_{n-1}, \cdots, u_{(n-k+1)}] \\ \hat{g} &= \left.\frac{\partial F_{actual}}{\partial u_n}\right|_{(y_n, \cdots, y_{n-k+1}; u_n=0, u_{n-1}, \cdots, u_{n-k+1})}\end{aligned} \tag{2.21}$$

As can be observed in (2.20), the estimated functions $\hat{f}$ and $\hat{g}$, which correspond to the submodels of the NARMA-L2 model, must be determined. The functions $\hat{f}$ and $\hat{g}$ in the NARMA-L2 model take into account both the previous output, y, and the control input, u. There is a linear relationship between the current control input, u_n, and the next output value, y_{n+d}. The task of solving the system identification problem for the system described above is essentially about determining the precise input-output mapping while minimizing the error as much as possible. To derive the NARMA-L2 model, we need to express equation (2.19) as a Taylor series expansion and then preserve only the first-order terms. As a result, the NARMA-L2 model $\hat{y}$ can be established as :

$$\hat{y}_{n+d} = \hat{f}(x_n) + \hat{g}(x_n)u_n \tag{2.22}$$

It should be noted that the input vector is x_n, and the functions $\hat{f}$ and $\hat{g}$ are nonlinear [1]. The control input u_n and the nonlinear dynamics are decoupled by the NARMA-L2 model, resulting in a linear form of the control input. The primary goal of this thesis is to increase the range of applications for inverse optimal controllers, particularly for non-affine systems. To do this, the non-affine NARX model of the system with ($LSSVR_{NARX}$), (ANN_{NARX}), and the input-output data are created. The built model is then used to calculate the NARMA-L2 submodels ($ANN_{NARMA-L2}$), ($LSSVR_{NARMA-L2}$). The NARMA-L2 model is illustrated in Fig. (5.2).

3. INVERSE OPTIMAL CONTROL

The main concept underlying inverse optimal control theory is to calculate a feedback control law that first achieves stabilization. It is a robust framework applied to solve the inverse problem in fields such as control systems, robotics, machine learning, and optimization. This approach considers optimal behaviors as a crucial factor in its application. Finding effective approaches to achieve optimal control of nonlinear non-affine systems remains a challenging problem in the field of control engineering. The complexity and difficulty in solving the corresponding Hamilton-Jacobi-Bellman (HJB) equations arise because particularly the system dynamics are not explicitly known which means there are no feasible solutions. Consequently, the research in this area is currently insufficient [98, 99]. As a result, in our research, we expanded the inverse optimal control's application area to non-affine systems.

The main idea behind inverse optimal control theory is the computation of a feedback control law that provides stabilization of the system as a first step. Then this control law is used to optimize a meaningful cost function by using the information on both the state variables and the control inputs [100–105]. Inverse optimal control theory might seem confusing when compared to the optimal control problem in which the cost functional needs to be pre-determined in order to construct a stabilizing control law. Inverse optimal control does not solve the problem of the HJB. Instead, it directly calculates the optimal control law by solving the Bellman problem. To solve the Bellman problem, the inverse control approach uses a candidate control Lyapunov function (CLF). A storage function is used as a candidate for this CLF. The output feedback control (or feedback control) is selected as the inverse optimum control law that the Bellman problem solves. The key concepts on inverse optimal control notions are summarized below [105–107]:

Due to the complexity of the solution of equation (2.8), researchers have introduced

the inverse optimal control technique as a viable alternative approach to overcome this difficulty.

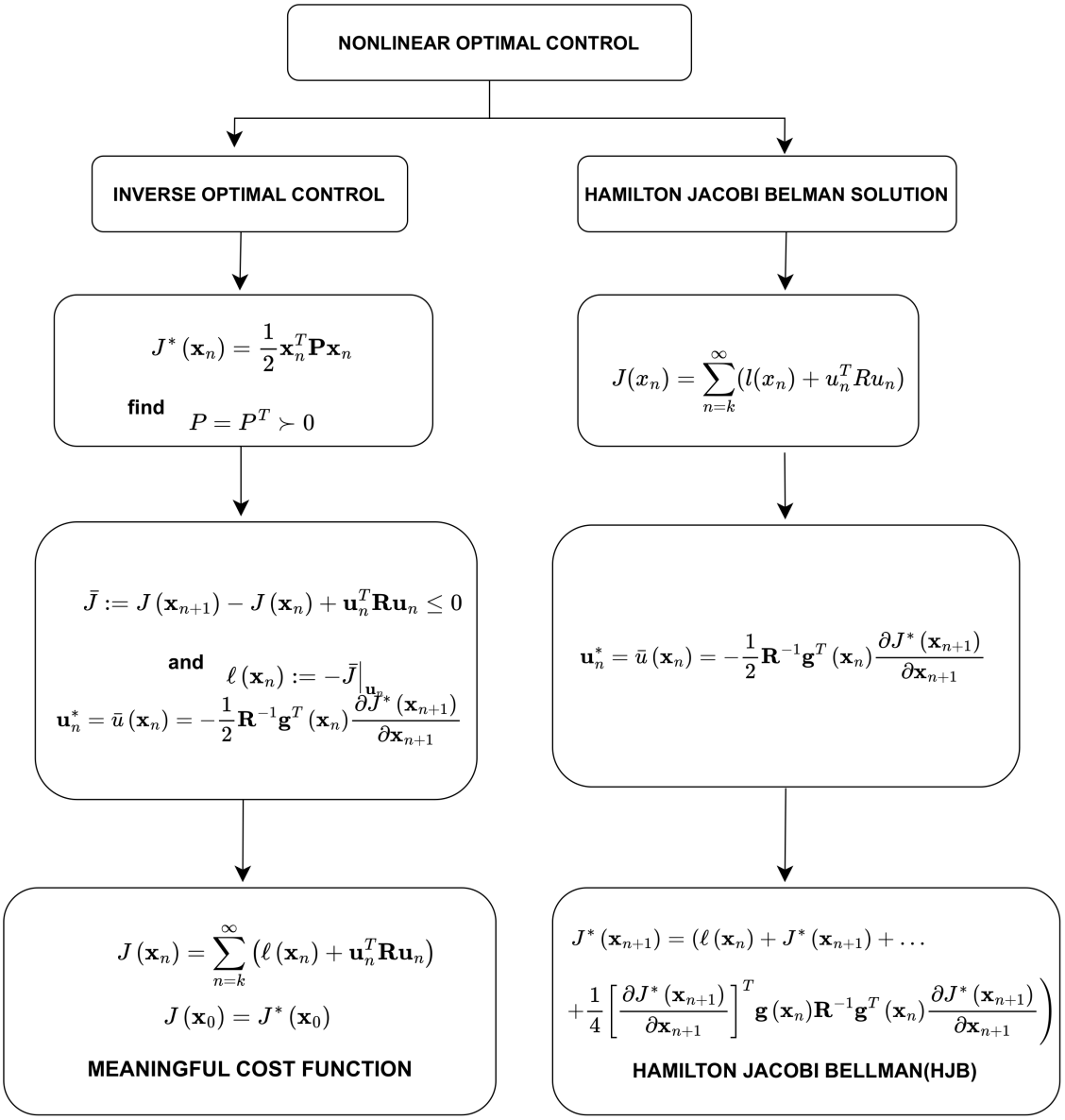

Figure 3.1 : A comparison of Inverse optimal control approach and traditional approach for optimal control problem.

Figure 3.1 visually highlights the difference between the conventional approach to solving the nonlinear optimal control problem and the inverse optimal control method grounded in the utilization of control Lyapunov functions.

• In the context of optimal control, $\ell(x_n)$ is predefined with $\ell(x_n) > 0$, along with the

input weighing term $R > 0$. Subsequently, these values are employed to compute $u^*(x_n)$ and $J^*(x_n)$ by solving the discrete-time Hamilton-Jacobi-Bellman (HJB) equation [58, 104, 106, 108].

• In the context of inverse optimal control, the control Lyapunov function $J(x_n)$ with a positive input weighting term $R > 0$ is provided initially. Subsequently, these functions are utilized to calculate $u^*(x_n)$ and the penalty term $\ell(x_n)$, which is used in the derivation of a meaningful cost function. This was accomplished by employing a stabilizing optimal controller, utilizing both Control Lyapunov Function (CLF) and passivity methodologies:

$$\begin{aligned} V(x_n) &= \min_u \sum_{n=0}^{\infty} l(x_n) + u^T R u \\ x_{n+1} &= f(x_n) + g(x_n)\, u_n \\ y_n = g(x_n)^T P f(x_n) &+ \frac{1}{2} g(x_n)^T P g(x_n) u_n \\ V(x_{n+1}) - V(x_n) &\leq y_n^T u_n \\ l(x_n) &\geq 0 \end{aligned} \tag{3.1}$$

where the passivity condition is $V(x_{n+1}) - V(x_n) \leq y_n^T u_n$ and the function $l(x_n)$ is a positive semidefinite function. The discrete-time Hamiltonian is first generated as $H(x_n, u_n) = l(x_n) + u^T R u + V(x_{n+1}) - V(x_n)$ in order to compute the problem's solution. Upon supposing that $V(x) = x^T P x$ has a positive definite matrix P, the control law can be obtained as follows: $u_n = -(I + \frac{1}{2} g^T(x_n))^{-1} g^T(x_n) P f(x_n)$. This yields $l(x_n) = -(f^T(x_n) P f(x_n) - x_n^T P x_n)$. Since $l(x_n) \geq 0$, the application of the passivity condition results in $u_n = -(I + \frac{1}{2} g^T(x_n) P g(x_n))^{-1} g^T(x_n) P f(x_n)$, a passivity-based feedback control law. Hence, it is now possible to compute the inverse optimal control law and generate the matrix P by using the state and control data in the demonstration trajectories.

Theorem 3.1 [58]: The following expression represents the optimal control law for trajectory tracking :

$$u_{c_n} = -\frac{1}{2} R^{-1} g^T(x) \frac{\partial V(z_{n+1})}{\partial z_{n+1}} \tag{3.2}$$

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

If the following two conditions are met, (3.2) is considered as an inverse optimal control input that globally stabilizes the system along $x_{\delta,n}$:

Condition 1) For the system given by (2.1), equation (3.2) satisfies (global) asymptotic stability of $x_n = 0$

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

$$\overline{V} := V(z_{n+1}) - V(z_n) + u_{c_n}^T R(z_n) u_{c_n} \leq 0 \tag{3.3}$$

If $l(z_n) := -\overline{V}$, then $V(z_n)$ is a solution for (2.8), and as a result, the cost functional associated with the tracking error is minimized. In order to calculate the inverse optimal control law for trajectory tracking of nonlinear systems, the first step is to choose a Control Lyapunov Function (CLF) that satisfies the required properties [58] as:

$$V(z_n) = \frac{1}{2} z_n^T P z_n \qquad\qquad P = P^T \succ 0 \tag{3.4}$$

in which trajectory tracking error (z_n) is shown as:

$$z_n = x_n - x_{\delta,n} = \begin{bmatrix} (x_{1,n} - x_{1\delta,n}) \\ \vdots \\ (x_{n,n} - x_{n\delta,n}) \end{bmatrix} \tag{3.5}$$

Consequently, the following equation can be calculated as the inverse optimal control law:

$$\begin{aligned} u_{c_n} &= \mid -\frac{1}{4} R^{-1} g^T(x_n) \frac{\partial z_{n+1}^T P z_{n+1}}{\partial z_{n+1}} \mid \\ &= \mid -\frac{1}{2} (R + P_2(x_n))^{-1} P_1(x_n, x_{\delta,n}) \mid \end{aligned} \tag{3.6}$$

where

$$P_1(x_n, x_{\delta,n}) = \begin{cases} g^T(x_n) P(f(x_n) - x_{\delta n+1}) \\ for f(x_n) \succeq x_{\delta,n+1} \\ g^T(x_n) P(x_{\delta,n+1} - f(x_n)) \\ for f(x_n) \preceq x_{\delta,n+1} \end{cases} \tag{3.7}$$

and

$$P_2(x_n) = \frac{1}{2} g^T(x_n) P g(x_n) \tag{3.8}$$

In (3.6), (3.7), (3.8), the selected P matrix is required to be positive definite and symmetric. The full proof of the theorem provided above is accessible through [58, 104, 106, 108]. The theorem mentioned before establishes the inverse optimal control law for trajectory tracking problems which can be computed by the right choice of P and R matrices for discrete-time affine-in-input nonlinear system models. In studies other than [109] the R value was taken constant throughout. In this thesis, the problem of reference tracking for discrete-time nonlinear systems with an input-non-affine structure has been transformed into the problem of finding an appropriate CLF, or in other words, the P and R matrix, just like in the regulator problem.

While the inverse optimal control technique has been widely applied for affine nonlinear systems in the technical literature, there is a lack of studies that focus on its application to non-affine systems. In this research, a novel methodology is proposed to convert a non-affine nonlinear system to an affine system which involves applying the NARMA-L2 modeling technique. Consequently, an inverse optimal controller can be constructed for the converted affine system.

In order to reduce the steady-state error that was noticed for the adaptive LSSVR-based inverse optimal controller in this study when we employed the inverse optimal controller alone, we additionally combined an integrator with the inverse optimal controller. By employing an integral control term, we reduced the steady-state error. As a result, the general control law used in this study can be expressed by the following equation:

$$u_{c_n} = | -\frac{1}{2}(R+P_2(x_n))^{-1}P_1(x_n,x_{\delta,n}) | + K_I \left[\sum_{i=1}^{k} z_i \right] \tag{3.9}$$

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

4. INVERSE OPTIMAL CONTROLLER DESIGN BASED ON NEURAL NETWORKS FOR NON-AFFINE SYSTEMS

4.1 Backpropagation Neural Networks

The term "Back Propagation" (BP) encompasses a diverse category of Artificial Neural Networks (ANNs) characterized by their architecture, which comprises various interconnected layers. The BP ANNs are a type of artificial neural network that employs a learning algorithm rooted in the steepest-descent technique. Given a suitable quantity of hidden units, these networks can effectively minimize the error of highly complex nonlinear functions. In theory, a BP network with a single layer of hidden units possesses the capability of mapping any nonlinear function $y = f(x)$.

The concentration of this thesis will be directed towards a singular neural network model: multilayered, feed-forward networks utilizing backpropagation.
In multilayer networks, the output from one layer serves as the input for the subsequent layer. The equations governing this process are as follows:

$$a^{m+1} = f^{m+1}\left(W^{m+1}a^m + b^{m+1}\right) \text{ for } m = 0, 1, \ldots, M-1 \tag{4.1}$$

Here, M denotes the number of layers in the network, W is the matrix of weights of the neural network, b represents the bias, a stands for the output of the neuron, and f is the activation function. The external inputs are the inputs to the neurons in the first layer:

$$a^0 = p, \tag{4.2}$$

This serves as the initial step for (4.1). The outputs produced by the neurons in the final layer are recognized as the network's outputs.

$$a = a^M, \tag{4.3}$$

The backpropagation algorithm for multilayer networks is generalized from the least means squares (LMS) algorithm, and both methods employ an identical performance index: mean square error. The algorithm is presented with a collection of instances demonstrating the desired behavior of the network:

$$\{p_1,t_1\},\{p_2,t_2\},\ldots,\{p_Q,t_Q\} \tag{4.4}$$

where p_q is an input to the network, and t_q is the corresponding target output. Upon applying each input to the network, a comparison is made between the network's output and the target. The algorithm must fine-tune the network parameters to minimize the mean square error:

$$F(x) = E\left[e^2\right] = E\left[(t-a)^2\right]. \tag{4.5}$$

Here, x denotes the vector encompassing the network's weights and biases. If the network yields multiple outputs, this concept can be extended to:

$$F(x) = E\left[e^T e\right] = E\left[(t-a)^T(t-a)\right] \tag{4.6}$$

Similar to the LMS algorithm, the mean square error is estimated using:

$$\hat{F}(x) = (t(k)-a(k))^T(t(k)-a(k)) = e^T(k)e(k) \tag{4.7}$$

Here, the squared error at the current iteration k takes the place of the expected squared error. The steepest descent algorithm for the approximate mean square error is expressed as:

$$w^m_{i,j}(k+1) = w^m_{i,j}(k) - \alpha\frac{\partial\hat{F}}{\partial w^m_{i,j}}, \tag{4.8}$$

$$b^m_i(k+1) = b^m_i(k) - \alpha\frac{\partial\hat{F}}{\partial b^m_i}, \tag{4.9}$$

where α is the learning rate.

Given that the error is an implicit function of the weights in the hidden layers, the chain rule of calculus is employed to determine the derivatives. To illustrate the chain

rule, consider a function f that solely depends on the variable n. We want to take the derivative of it with respect to a third variable w. The chain rule is then:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw}. \tag{4.10}$$

This concept is applied to compute the derivatives in (4.8) and (4.9).

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \tag{4.11}$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \tag{4.12}$$

The second term in these equations can be calculated in a straightforward way, as the net input to layer m is a direct function of the weights and bias in that layer:

$$n_i^m = \sum_{j=1}^{S^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \tag{4.13}$$

Hence,

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \frac{\partial n_i^m}{\partial b_i^m} = 1 \tag{4.14}$$

If we now define sensitivity of $\hat{F}$

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m}, \tag{4.15}$$

then (4.11) and (4.12) can be simplified to

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1}, \tag{4.16}$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \tag{4.17}$$

The approximate steepest-descent algorithm can now be represented as:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1}, \tag{4.18}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m. \tag{4.19}$$

This can be represented in matrix form as:

$$W^m(k+1) = W^m(k) - \alpha s^m \left(a^{m-1}\right)^T \tag{4.20}$$

$$b^m(k+1) = b^m(k) - \alpha s^m \tag{4.21}$$

4.2 The Proposed Inverse Optimal Controller Design Based on Neural Network For Nonlinear-nonaffine Systems

This approach involves transforming a NARX model into a NARMA-L2 model, enabling the representation of an initially non-affine system as an affine model by using neural networks with backpropagation algorithm. As a result, a system that was initially non-affine can now be expressed as an affine model. The entire process comprises two steps. Initially, the input-output data of the system to be controlled is extracted. Subsequently, a feedforward neural network is employed to derive the NARMA-L2 model of the system based on this collected dataset. Upon gathering input-output data from the plant, this dataset embodies the NARX model of the system, characterized as non-affine. Through the process of converting from the NARX model to the NARMA-L2 model, an affine representation is achieved, effectively transforming it from its original non-affine model.

Figure 4.1 given below illustrates the feedforward neural network utilized in the transition from the NARX model to the NARMA-L2 model:

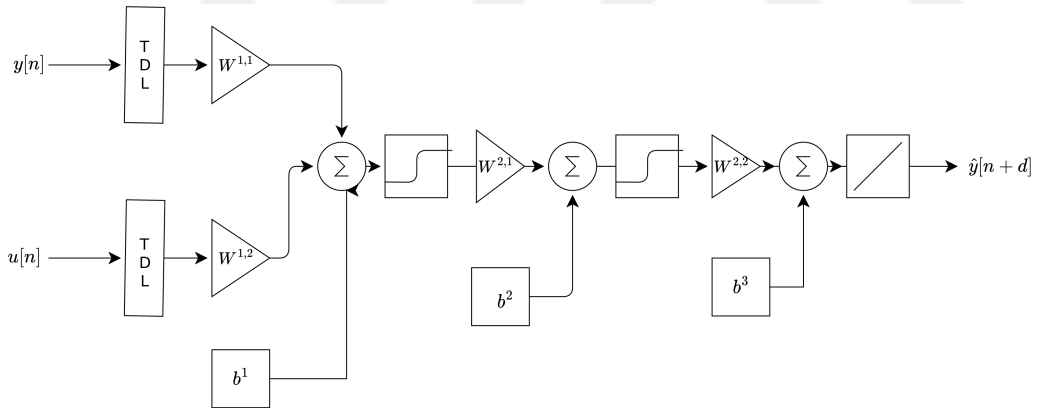

Figure 4.1 : NARMA-L2 neural network structure for system identification

The neural network receives as input the data comprising both input and output values from the plant to be modeled. In the initial layer, the input and output data undergo multiplication with distinct sets of weights. Following this, they are passed through an activation function after incorporating a bias value. The neural network constructs the submodel of the NARMA-L2 model. Distinct neural networks are used to independently estimate $\hat{f}(.)$ and $\hat{g}(.)$. The NARMA-L2 model identification process

is conducted offline. Once the identification of the plant for control is finished, the subsequent phase involves the adaptive online implementation of the inverse optimal control methodology. The formulation of the inverse optimal control law has been provided in equations (3.2) and (3.6) in section 3. The P matrix in this equation plays a crucial role, as its value directly impacts the performance of the control system. As a result, numerous approaches have been suggested in existing literature for optimizing the P matrix [110–112]. This thesis introduces an innovative method in which the P matrix is adaptively determined for optimal results by using neural networks. An online operating recurrent neural network continuously calculates the optimal values for the components of the P matrix, aiming to minimize a metric based on the control error.

Figure 4.2 depicts the schematic diagram of the suggested control architecture.

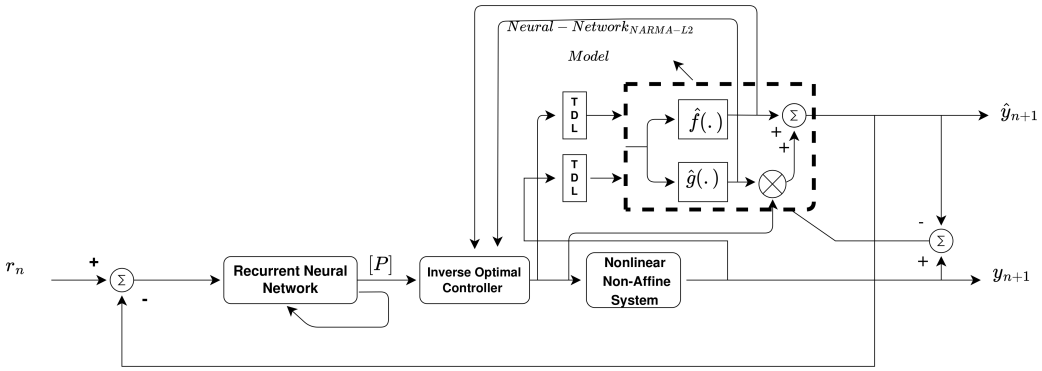

Figure 4.2 : Proposed inverse optimal control structure for non-affine nonlinear systems

In this design, the NARMA-L2 sub-modules $\hat{f}(.)$ and $\hat{g}(.)$, determined through the feedforward neural network, are employed to calculate the anticipated system output y_{n+d}. To determine the entries of the P matrix, a recurrent neural network is utilized, as illustrated below:

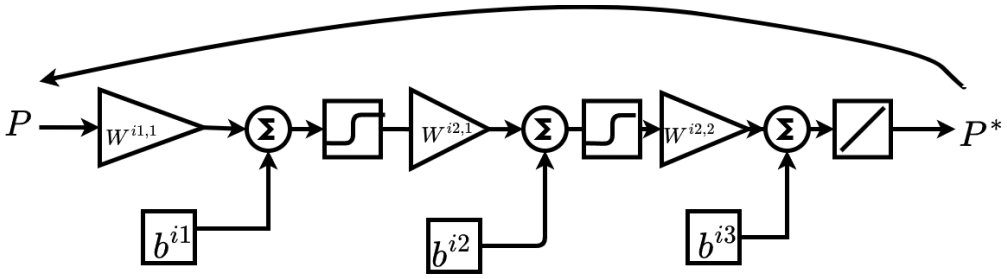

Figure 4.3 : Recurrent neural network structure for the calculation of controller parameters

The recurrent neural network calculates the optimal P^* matrix as follows:

$$P^* = {W_n^i}^T P W_n^i \tag{4.22}$$

The weights are adjusted to minimize a discrete Lyapunov function.

For trajectory tracking, the Lyapunov function is formulated as:

$$V^*(z_n) = \frac{1}{2} z_n^T P^* z_n \tag{4.23}$$

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

$$e_{t_n} = r_n - \hat{y}_{n+d} \tag{4.24}$$

Thus, for trajectory tracking, the weights are updated based on the update rule outlined below:

$$W_{n+1}^i = W_n^i - \alpha P \sum_{k=1}^{N} \mid \frac{e_t}{n} \mid \tag{4.25}$$

As a result, the components of the P matrix undergo an update. The sequential processes of obtaining the non-affine NARMA-L2 model of the system using a feedforward neural network and computing the optimal P^* matrix using the recurrent neural network is outlined below:

STEP1 – OFFLINE SYSTEM IDENTIFICATION

1. Set the controller and model parameters at the beginning.
-P, R, W^i, W, n_u, n_y.
- Input control signal u_n is to be constructed. To compute system output y_n, apply this signal to the plant that has to be controlled.

2. - To gather a training dataset, generate $\begin{bmatrix} u_n & y_n \end{bmatrix}^T$ by utilizing the produced input-output pairs in Step 1.
-Here

$$\begin{aligned} u_n &= \begin{bmatrix} u_n & u_{n-1} & u_{n-2} & u_{n-3} \end{bmatrix} \\ y_n &= \begin{bmatrix} y_n & y_{n-1} & y_{n-2} & y_{n-3} \end{bmatrix} \end{aligned} \tag{4.26}$$

3. Build feedforward neural networks in the manner described above, then train them to estimate NARMA-L2 submodels $\hat{f}(.)$ and $\hat{g}(.)$ using the dataset generated in Step 2. The method employs distinct feedforward neural networks to estimate $\hat{f}(.)$ and $\hat{g}(.)$.

STEP2 –ADAPTIVE ONLINE CONTROL

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

$$\widehat{y}_{n+d} = \widehat{f}(x_n) + \widehat{g}(x_n)u_n \tag{4.27}$$

-This will serve as the source of data regarding the real system output.

2. Determine the error $e_t[n]$. The difference between the estimated system output and the reference signal is established for the trajectory tracking problem.

$$e_{t_n} = r_n - \widehat{y}_{n+d} \tag{4.28}$$

3. To minimize the error, update the weights (W^i) of the designed recurrent neural network, which was previously explained in detail, using the error supplied by the problem:

$$W^i_{n+1} = W^i_n - \alpha P \sum_{k=1}^{N} \left| \frac{e_n}{N} \right| \tag{4.29}$$

4. The optimal P matrix, P^*, is the recurrent neural network's output. Utilize the calculated weights to determine the matrix's elements, P^*:

$$P^* = W^{i^T}_{n+1} P W^i_{n+1} \tag{4.30}$$

5. Find the inverse optimal control law using the P matrix that was obtained.

$$u_n{}^* = \alpha(x_n) = -\frac{1}{2}\left(R(x_n) + P_2(x_n)\right)^{-1} P_1(x_n) \tag{4.31}$$

-where

$$P_1(\boldsymbol{x}_n) = \hat{g}^T(\boldsymbol{x}_n) P^* \hat{f}(\boldsymbol{x}_n) \tag{4.32}$$

and

$$P_2(\boldsymbol{x}_n) = \hat{g}^T(\boldsymbol{x}_n) P^* \hat{g}(\boldsymbol{x}_n) \tag{4.33}$$

6. Use the calculated control input signal on the system that is expected to be controlled.

4.3 Simulation Results For Neural Network Based Adaptive Inverse Optimal Controller For Nonlinear Non-Affine Systems

The efficacy of the proposed neural network-based conversion method, which transforms a non-affine system model into an affine one using the NARMA-L2 model, along with an adaptive online inverse optimal control approach, has been validated through extensive simulations on two distinct benchmark systems. These simulations are conducted under two additional conditions to assess robustness: firstly, introducing a sinusoidal disturbance with an amplitude of 2 units, and secondly, applying white noise with a signal-to-noise ratio (SNR) of 2 dB to the system. It is assumed that both the disturbance and noise persistently influence the system throughout the simulation.

Consequently, the simulation results are presented for the following cases:

1) Trajectory tracking performance under conditions with no disturbance or noise.
2) Trajectory tracking performance when a sinusoidal disturbance with a magnitude of 2 units is applied.
3) Trajectory tracking performance when white noise with an SNR of 2 dB is applied.

4.3.1 Benchmark system I

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

$$\begin{aligned} y_{n+1} = {} & 0.2cos(0.8y_n + y_{n-1}) + 0.4sin(0.8(y_n + y_{n-1}) + 2u_n + u_{n-1} \\ & + 0.1(9 + y_n + y_{n-1}) + \frac{2(u_n + u_{n-1})}{10 + cos(y_n)}) \end{aligned} \tag{4.34}$$

The initial condition is given as $y(0) = 0$. The system output graphs for different scenarios, the P parameter, the trajectory tracking error, and the derived inverse optimal control law are shown in figures below.

4.3.1.1 Nominal case

Figure 4.4 illustrates the performance of the proposed method in addressing the trajectory tracking problem. It is evident that the proposed control methodology effectively achieves successful tracking while rapidly reducing transient effects.

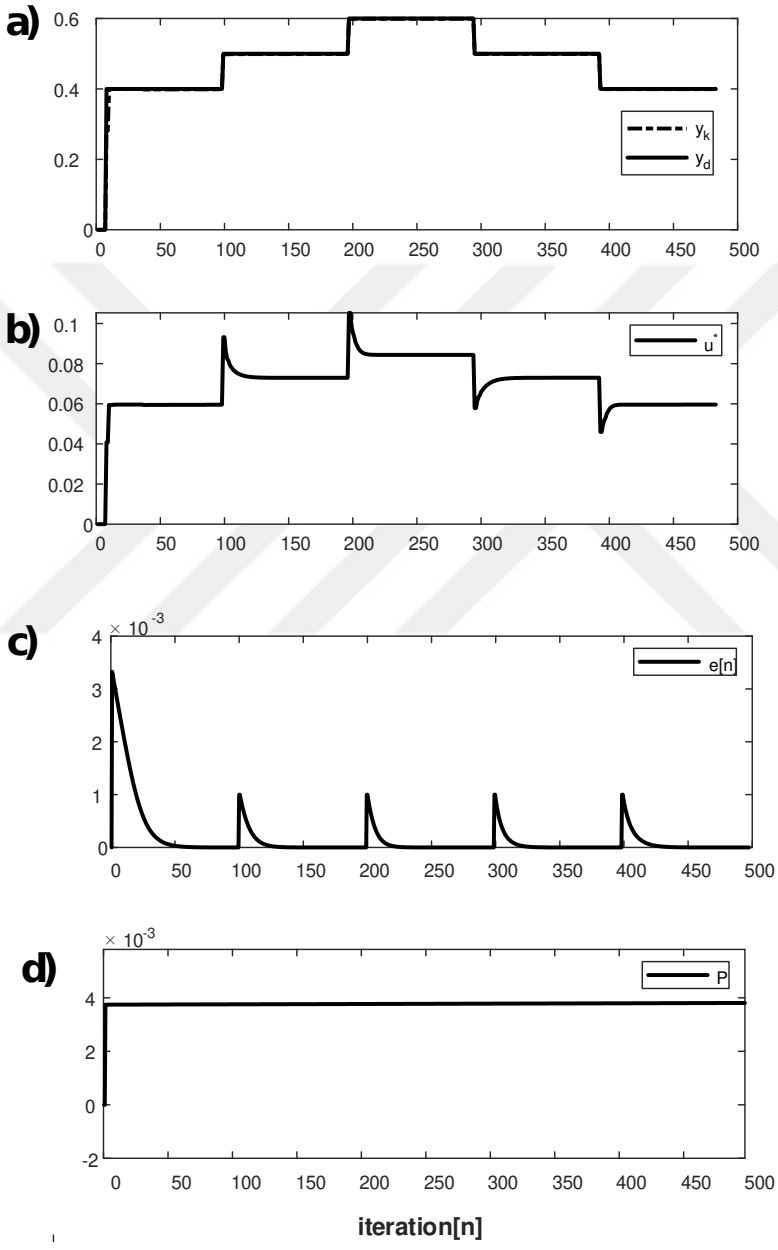

Figure 4.4 : Nominal case trajectory tracking results for benchmark system I.

4.3.1.2 Input disturbance case

Figure 4.5 illustrates the trajectory tracking performance of the system when a sinusoidal disturbance with a magnitude of 2 units is applied. It can be clearly seen that the oscillatory effects are surpassed and the controller's parameter is fine-tuned by the proposed control structure.

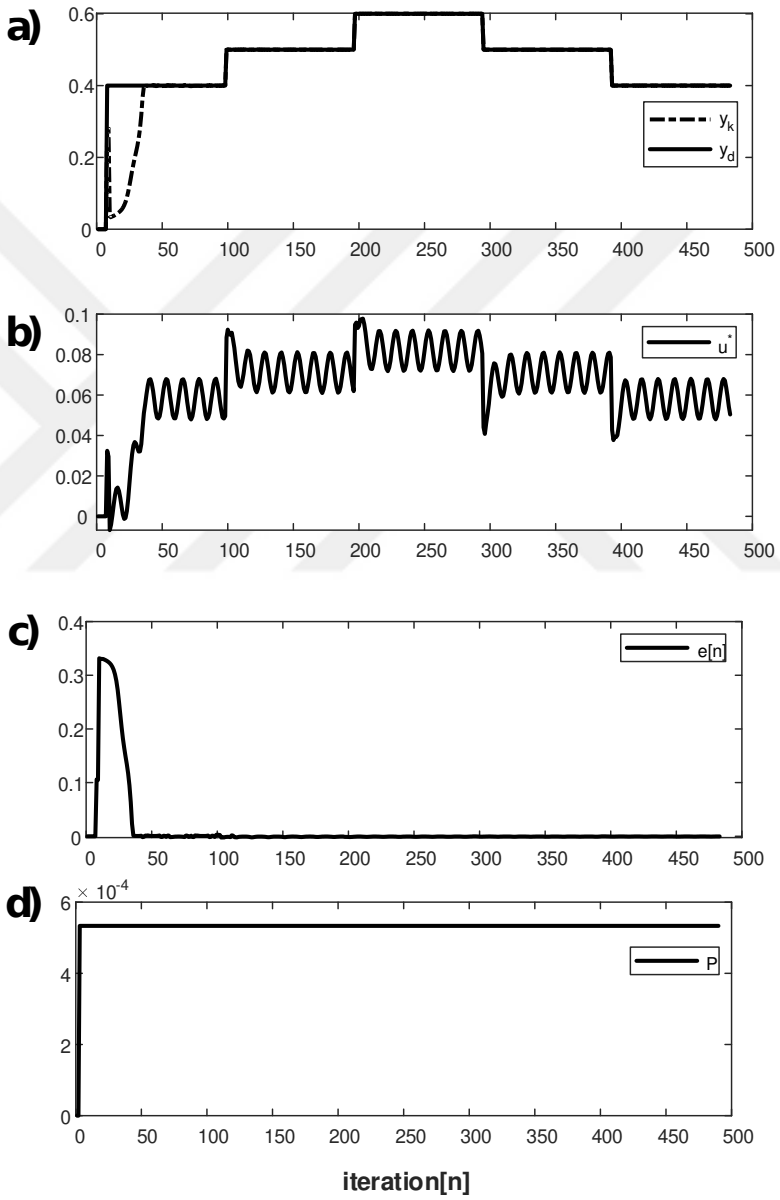

Figure 4.5 : Trajectory tracking results when sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system I.

4.3.1.3 Measurement noise case

Figure 4.6 depicts the case results when white noise with a SNR of 2 DB is added to the system. The computed control input and the controller parameter are also illustrated, it clearly shows that the proposed control method can successfully provide control and also can effectively deal with the noise.

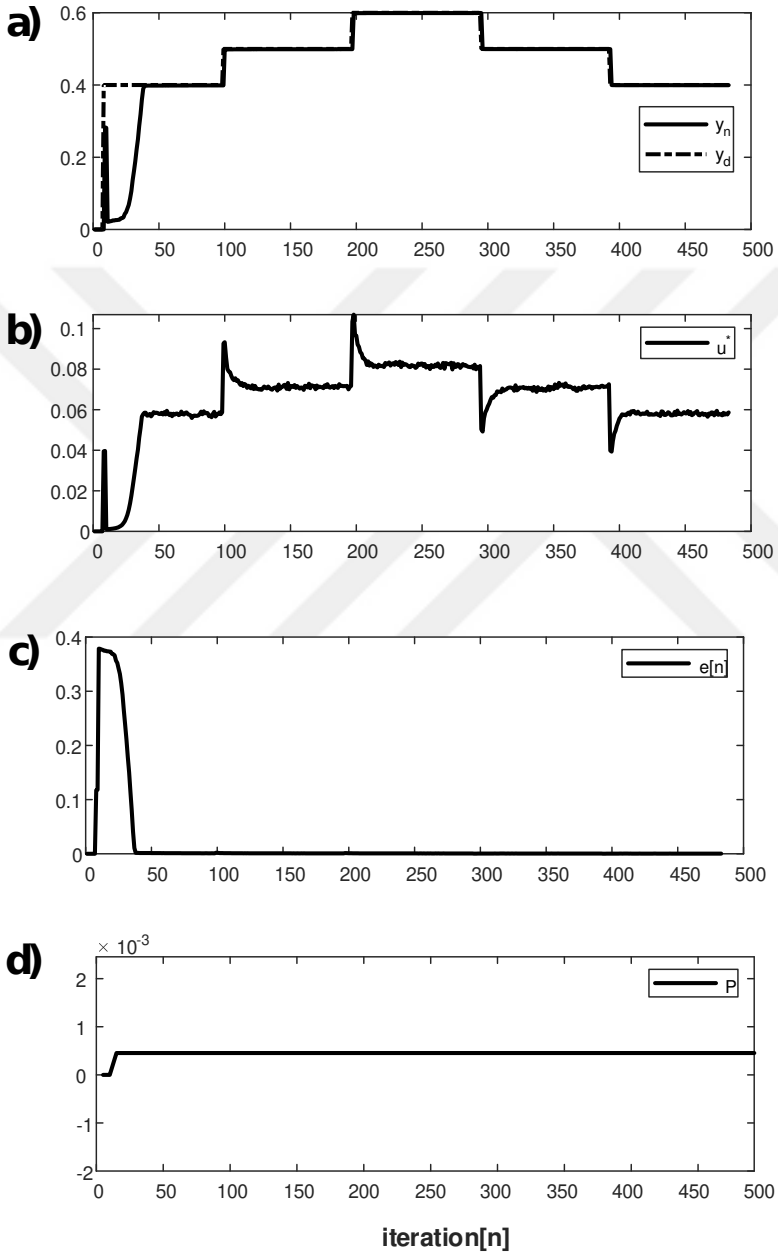

Figure 4.6 : Trajectory tracking results with 2 dB (SNR) white noise for benchmark system I.

4.3.2 Benchmark system II

The second benchmark system is a second order non-affine plant that is characterized by the equations given in (4.35).

$$\begin{aligned} x_{1_{n+1}} &= x_{2_n} \\ x_{2_{n+1}} &= sin(x_{1_n} u_n) + u_n(5 + cos(u_n x_{2_n})) \\ y_n &= x_{1_n} \end{aligned} \tag{4.35}$$

Figures 4.7 to 4.9 display the trajectory tracking results for benchmark system II. These figures include graphs depicting the system output, the computed inverse optimal control law, the trajectory tracking error, and the P parameter.

4.3.2.1 Nominal case

Figure 4.7 presents the trajectory tracking results under nominal conditions, where there is no disturbance or noise present.

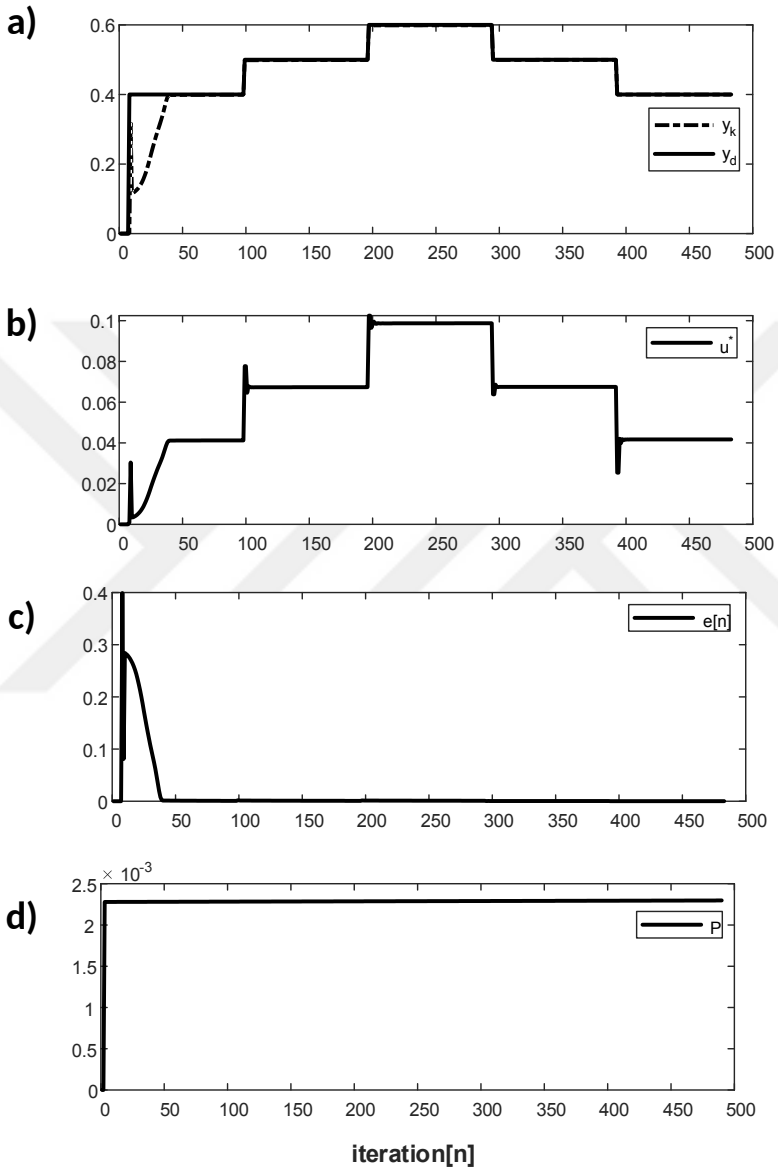

Figure 4.7 : The nominal case tajectory tracking results for benchmark system II.

4.3.2.2 Input disturbance case

In Figure 4.8, trajectory tracking results, the generated control input and the controller parameter obtained for the case when a sinusoidal disturbance with a magnitude of 2 units is applied to the system are shown. The success of the proposed control method is clearly observed.

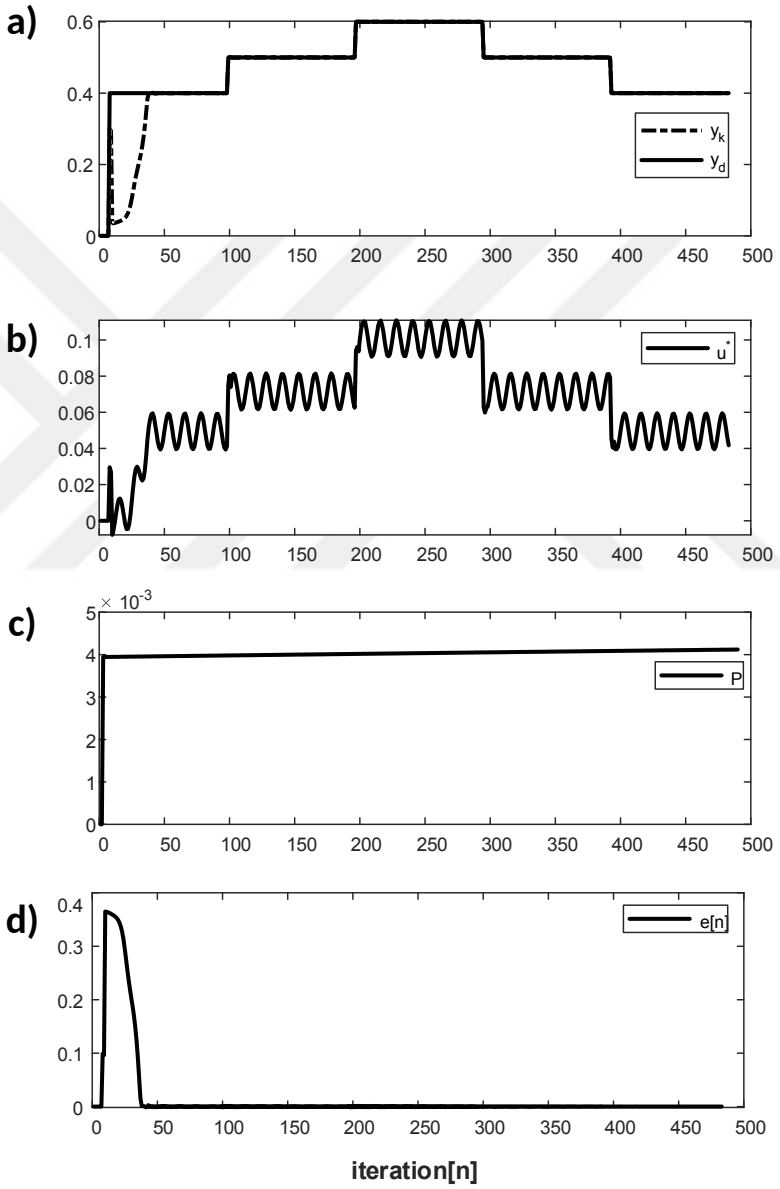

Figure 4.8 : Trajectory tracking results when a sinusoidal disturbance with a magnitude of 2 units is applied for benchmark system II.

4.3.2.3 Measurement noise case

Figure 4.9 depicts the simulation results when a 2dB white noise is applied to the system. The outcomes validate that the suggested control strategy successfully performs accurate trajectory control and effectively mitigates the impact of measurement noise.

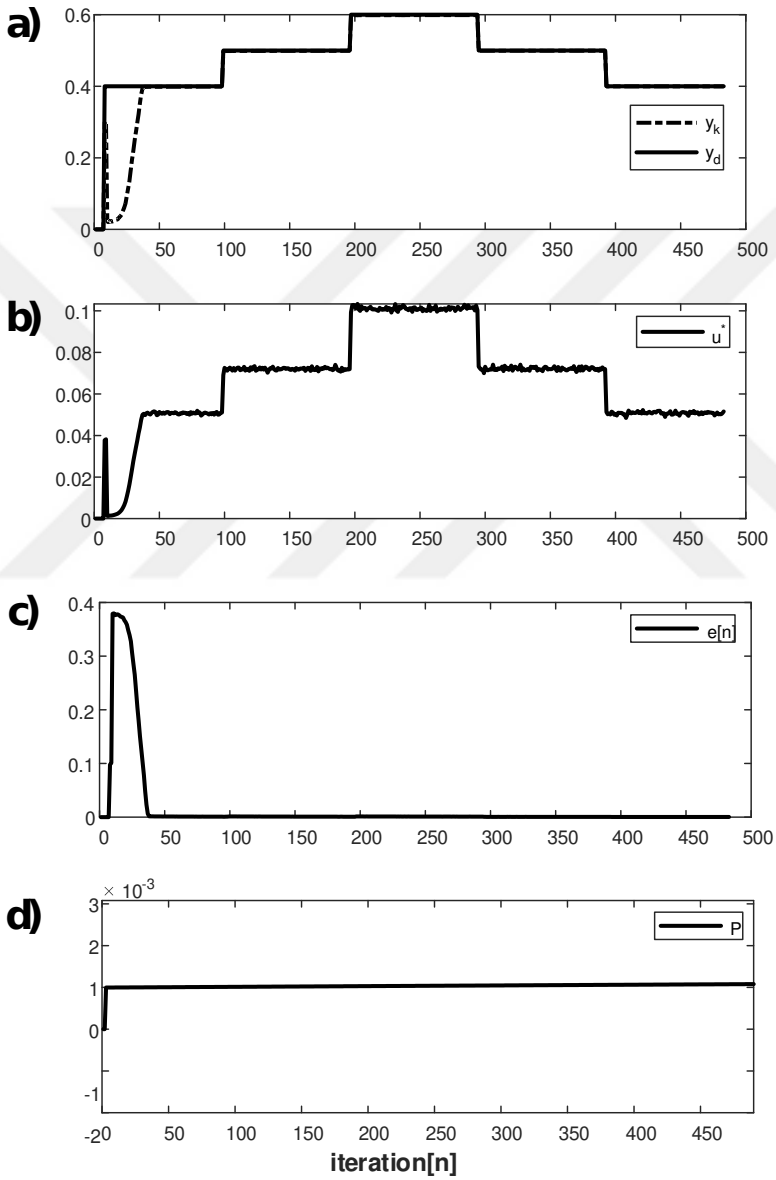

Figure 4.9 : Trajectory tracking results with 2 dB (SNR) white noise for benchmark system II.

5. ADAPTIVE INVERSE OPTIMAL CONTROLLER BASED ON LEAST SQUARES SUPPORT VECTOR REGRESSION FOR NON-AFFINE SYSTEMS

5.1 Least Squares Support Vector Regression For Nonlinear Function Estimation

LSSVR is a popular regression approach that has its origins in the conventional SVR approach [113], the key characteristics of LSSVR are its use of equality constraints and a squared error term in the optimization problem formulation.
A training set:

$$(y_1,\underline{x_1}),\cdots,(y_k,\underline{x_k}),x\in R^n,y\in R,k=1,2,\cdots,N \tag{5.1}$$

is given where N is the size of training data and n is the dimension of the input matrix. The optimization problem is expressed as:

$$\underset{w,b,e}{min}\frac{1}{2}w^T w+\frac{1}{2}\sum_{j=1}^{N}e_k^2 \tag{5.2}$$

subject to equality constraints:

$$y_k-w^T\varphi(x_k)-b-e_k=0,k=1,2,\cdots,N \tag{5.3}$$

The equations (5.2) and (5.3) are denoted as the primal form in the context of the minimization problem. The Lagrange function of the LSSVRs is formulated out of the primary objective function and its accompanying constraints to a dual set of variables [114]. The primal objective function and constraints can be used to obtain the Lagrangian function:

$$L(w,b,e,a)=\frac{1}{2}w^T w+\frac{1}{2}C\sum_{k=1}^{N}e_k^2-\sum_{k=1}^{N}\alpha_k(w^T\varphi(x_k)+b+e_k-y_k) \tag{5.4}$$

where $\varphi(x_k)$ denotes a nonlinear mapping that converts the input dataset to feature space, e_k is the error of least-squares data fitting, L represents the Lagrangian, C

is the penalty term, b stands for bias, a_k are Lagrange multipliers, and parameter w is the weight vector [114–116]. Considering that the Lagrangian exhibits saddle point behavior at the solution involving both the primal and dual variables, optimality requires that the partial derivatives of the Lagrangian with respect to the primal variables are zero. Using the optimality requirements of Karush-Kuhn-Tucker, the following results are obtained:

$$\frac{\partial L}{\partial b} = \sum_{k=1}^{N} \alpha_k = 0 \tag{5.5}$$

$$\frac{\partial L}{\partial w} = w - \sum_{k=1}^{N} \alpha_k \phi(x_k) = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_k \varphi(x_k) \tag{5.6}$$

$$\frac{\partial L}{\partial e_k} = C \sum_{k=1}^{N} e_k - \sum_{k=1}^{N} \alpha_k = 0 \rightarrow \alpha_k = C e_k \tag{5.7}$$

$$\frac{\partial L}{\partial \alpha_k} = 0 \rightarrow y_k = w^T \varphi(x_k) + b + e_k \tag{5.8}$$

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

$$\begin{bmatrix} 0 & 1 \\ 1^T & \Omega + \dfrac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ a^T \end{bmatrix} = \begin{bmatrix} 0 \\ y^T \end{bmatrix} \tag{5.9}$$

where $y = \left[y_1, y_2, \cdots y_N\right], a = \left[a_1, a_2 \cdots a_N\right], 1 = \left[1, 1 \cdots 1\right], \Omega = K(x_k, x_m), \quad k, m \in \{1, 2, \cdots N\}$. The nonlinear autoregressive with exogenous inputs (NARX) model can be obtained as:

$$y_n = f\left(u_n, .., u_{n-n_u}, y_{n-1}, .., y_{n-n_y}\right) \tag{5.10}$$

where y_n is the plant's output, n_u and n_y indicate the order of the NARX model, and u_n is the control input given to the plant at time n. The dynamics of the non-linear system can be embodied using this model. The state vector of the system at time index n is shown as follows:

$$x_n = \left[u_n, .., u_{n-n_u}, y_{n-1}, .., y_{n-n_y}\right] \tag{5.11}$$

Therefore, as shown in (5.12), the non-linear regression function can be formulated in terms of the support vectors:

$$f(x) = \sum_{k \in SV} a_k K(x_n, \underline{x_k}) + b \tag{5.12}$$

For more comprehensive and detailed information, please see the following additional resources [114–116].

5.2 NARMA-L2 modeling of Nonlinear Non-Affine Systems Based on Least Squares Support Vector Regression

The implementation of a novel technique to construct an inverse optimal controller for non-affine and nonlinear systems is one of the primary goals of this study. Consequently, the step of establishing the NARMA-L2 model holds significant importance in this context. In literature suggested adaptive controller architectures for non-affine systems rely on constructing the system model using the NARX model to predict future Jacobian information. The primary drawback of this structure is that the control signal becomes intertwined with nonlinear dynamics. Nonetheless, one of the key innovations in the proposed research centers on the conversion of the control signal term, originally concealed within the NARX model's nonlinear dynamics. This transformation results in a model known as NARMA, where it is isolated from the nonlinear dynamics and manifests as a multiplier term. To accomplish this, the process begins by obtaining the NARX model of the system as the first step. Subsequently, utilizing this acquired NARX model allows for the development of the NARMA-L2 model. So, in the initial step, the NARX model of the system is easily constructed, using the accessible input-output data. To effectively use an inverse optimal controller, it is essential to subdivide the dynamics of the NARX model into a NARMA-L2 model [1, 117–119].

The online LSSVR method has been utilized for two purposes. Firstly, it is used to acquire the NARX model, and subsequently, to transform it into the NARMA-L2 model. The two independent LSSVRs are formed as shown in (5.1). To derive a useful system model for implementing the inverse optimal controller, $LSSVR_{NARX}$ generates

the NARX model of the system using the input-output dataset of the system.

Next, $LSSVR_{NARMA-L2}$ subdivides this model into a NARMA-L2 model.

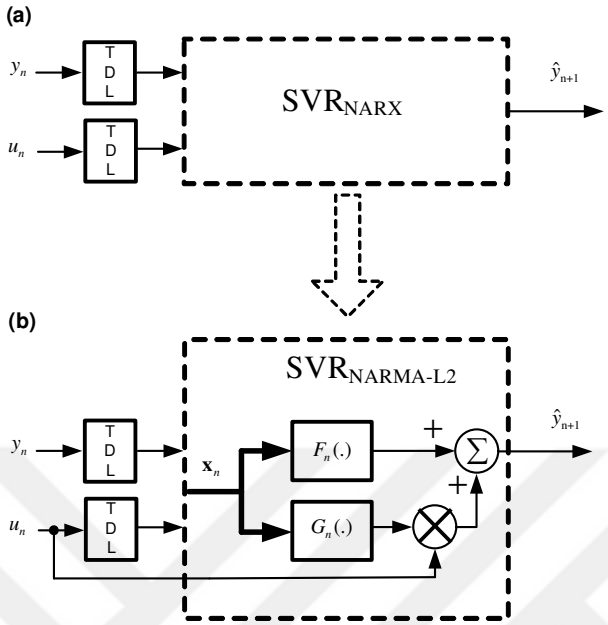

Figure 5.1 : Decomposition of $LSSVR_{NARX}$ model to $LSSVR_{NARMA-L2}$ model [1]

5.2.1 NARX system identification based least squares support vector regression

The NARMA-L2 model of the system to be controlled has been attained using the online *LSSVR* method. To achieve this objective, the initial step is to obtain the *NARX* model, which is then utilized to derive the NARMA-L2 model. The dynamics of a non-linear system can be depicted by the Nonlinear AutoRegressive with eXogenous inputs (*NARX*) model,

$$y_{n+1} = f(\left[u_n, \cdots, u_{n-n_u}, y_{n-1}, \cdots, y_{n-n_y}\right]) \tag{5.13}$$

where y_{n+1} is the plant's output, u_n is the control input given to the plant at time n, and n_u and n_y denote the number of previous control inputs and plant outputs included in the model, respectively [120]. The following is a representation of the system's state vector at time index n:

$$x_n = \left[u_n, \cdots, u_{n-n_u}, y_{n-1}, \cdots, y_{n-n_y}\right] \tag{5.14}$$

Using equations (5.12), (5.13) and (5.14), the output of the model can be written as:

$$\hat{y}_{NARX_{n+1}} = \sum_{i=n-L}^{n-1} a_i K(x_n, x_i) + b_n \tag{5.15}$$

Using the inputs $X_n = [x_{n-1}, \cdots, x_{n-L}]$, the system's corresponding outputs $Y_n = [y_n, \cdots, y_{n-L+1}]$, the sliding window length L and Lagrange multipliers a_i, the training data set (X_n, Y_n) may be determined. If we assume $U_n = [\Omega_n + \frac{I}{C}]^{-1}$ and use (5.9), a_n and b_n are obtained as:

$$Y_n = [U_n]^{-1} a_n^T + 1^T b_n \tag{5.16}$$

$$1 a_n^T = 0 \tag{5.17}$$

$$U_n Y_n = U_n [U_n]^{-1} a_n^T + U_n 1^T b_n \tag{5.18}$$

$$a_n^T = U_n [Y_n - 1^T b_n] \tag{5.19}$$

Using (5.18)

$$1 a_n^T = 1 U_n [Y_n - 1^T b_n] = 0 \tag{5.20}$$

$$b_n = \frac{1 U_n Y_n}{1 U_n 1^T} \tag{5.21}$$

Thus at time index n :

$$X_n = [x_{n-1}, \cdots, x_{n-L}] \tag{5.22}$$

$$Y_n = [y_n, \cdots, y_{n-L+1}] \tag{5.23}$$

At time index $n+1$:

$$X_{n+1} = [x_n, \cdots, x_{n-L+1}] \tag{5.24}$$

$$Y_{n+1} = [y_{n+1}, \cdots, y_{n-L+2}] \tag{5.25}$$

New data pair (x_n, y_n) gets introduced to the training data at time index $n+1$, and old data pair (x_{n-L}, y_{n-L+1}) is eliminated from the data set [121, 122].

5.2.2 NARMA-L2 decomposition based least squares support vector regression

After acquiring the system's NARX model, the next step is to partition it down into NARMA-L2 submodels. Therefore, the main goal of this step is to use the system model that was previously obtained to generate the model parameters for $LSSVR_{NARMA-L2}$.

$$\hat{y}_{NARX_{n+1}} = \sum_{i=n-L}^{n-1} a_i K(x_n, x_i) + b_n \tag{5.26}$$

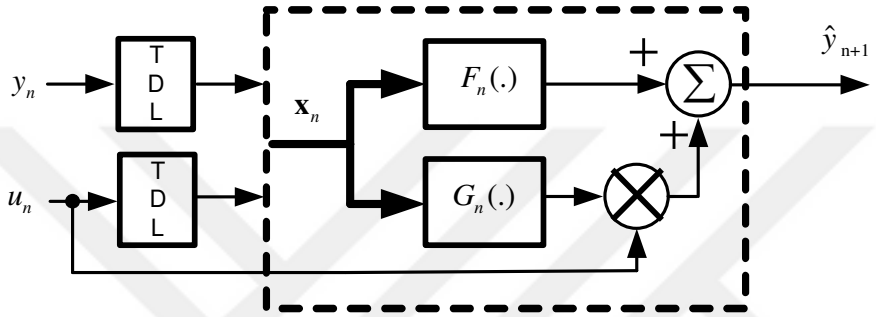

Figure 5.2 : $LSSVR_{NARMA-L2}$ model [2], [1]

In order to make the NARX model usable for inverse optimal control, it is necessary to decompose it into submodels denoted as $F_n(.)$ and $G_n(.)$, as depicted in Figure (5.2). These submodels can then be utilized during the prediction step:

$$\hat{f}_n^- \cong F_{n-1}(x_n) \tag{5.27}$$

$$\hat{g}_n^- \cong G_{n-1}(x_n) \tag{5.28}$$

The subscript indicates the time index of the state vector, while the superscript denotes whether the system model is derived for time step n or $n-1$. The "-" symbol is employed to indicate that models calculated in the previous time step $n-1$ are assessed using the current state vector x_n to determine $\hat{f}_n^-$ and $\hat{g}_n^-$. In simpler terms, $\hat{f}_n^+$ and $\hat{g}_n^+$ indicate that the model obtained at the current step (n) has been employed using the current state vector x_n as shown in the following:

$$\hat{f}_n^+ \cong F_n(x_n) \tag{5.29}$$

$$\hat{g}_n^+ \cong G_n(x_n) \tag{5.30}$$

To apply inverse optimal control, it is necessary to acquire the submodels $F_n(.)$ and $G_n(.)$. The following equations describe regression functions for $F_{n-1}(x_{n-1})$ and $G_{n-1}(x_{n-1})$:

$$\hat{f}_{n-1}^{+} \cong F_{n-1}(x_{n-1}) = \sum_{i=1}^{N} \alpha_i k(x_i, x_{n-1}) + b_f \tag{5.31}$$

$$\hat{g}_{n-1}^{+} \cong G_{n-1}(x_{n-1}) = \sum_{i=1}^{N} \theta_i k(x_i, x_{n-1}) + b_g \tag{5.32}$$

To obtain the output of the $LSSVR_{NARMA-L2}$ model of the system, the following approach has been used:

$$\hat{y}_{NARMA_n} = F_{n-1}(x_{n-1}) + G_{n-1} u_{n-1} \tag{5.33}$$

$$= \sum_{i=1}^{N} \alpha_i k(x_i, x_{n-1}) + b_f + \left[\sum_{i=1}^{N} \theta_i k(x_i, x_{n-1}) + b_g \right] u_{n-1} \tag{5.34}$$

When the models of $LSSVR_{NARMA-L2}$ and $LSSVR_{NARX}$ are compared, a correspondence can be formed between the two models:

$$\sum_{i=1}^{N} a_i K(x_i, x_{n-1}) + b_n = \sum_{i=1}^{N} [\alpha_i + \theta_i u_{n-1}] K(x_i, x_{n-1}) + b_f + b_g u_{n-1} \tag{5.35}$$

So

$$a_i = \alpha_i + \theta_i u_{n-1} \tag{5.36}$$

$$b_n = b_f + b_g u_{n-1} \tag{5.37}$$

To proceed further, we will assume the correspondence of a relationship between the submodels.

$$\hat{y}_{NARX_n} \cong \hat{y}_{NARMA_n} \tag{5.38}$$

and

$$\alpha_i = \mu_1(.)\theta_i \tag{5.39}$$

$$b_f = \mu_2(.) b_g \tag{5.40}$$

Therefore, finally, we acquire the following:

$$F_{n-1}(x_{n-1}) = \mu_1(.)G_{n-1}(x_{n-1}) + [\mu_2(.) - \mu_1(.)]b_g \tag{5.41}$$

The expression in equation (5.41) reveals that the $LSSVR_{NARMA-L2}$ model of the system is solely reliant on $G_{n-1}(x_{n-1})$ when the given values of regression parameters $\mu_1(.)$ and $\mu_2(.)$ are considered. The computation of the bias and Lagrange parameters for the $LSSVR_{NARMA-L2}$ submodels, $F_{n-1}(x_{n-1})$ and $G_{n-1}(x_{n-1})$, is accomplished by employing equations (5.39) and (5.40) according to the $LSSVR_{NARX}$ model as shown:

$$\begin{aligned} \theta_i &= \frac{a_i}{\mu_1(.) + u_{n-1}}, a_i = \mu_1(.)\theta_i \\ b_g &= \frac{b_n}{\mu_2(.) + u_{n-1}]}, b_f = \mu_2(.)b_g \end{aligned} \tag{5.42}$$

As a result, the essential system dynamics for developing an inverse optimal controller can be determined through computations based on the initially acquired $LSSVR_{NARX}$ model, subsequently, this model can be transformed into a $LSSVR_{NARMA-L2}$ model.

5.3 Online Adaptive Parameter Optimisation With Levenberg-Marquardt Algorithm

At every iteration the Levenberg-Marquardt algorithm is employed to fine-tune the controller and model parameters in this thesis study. The parameters μ_1, μ_2, P, R, and K_I significantly determines the tracking performance of the proposed method. The effective update of all the mentioned parameters is ensured through the implementation of a predictive framework that takes into consideration the system's behavior K steps ahead into the future. The Levenberg-Marquardt algorithm is employed to optimize the objective function defined in equation (5.43) and fine-tune the parameters P, R, μ_1, μ_2, K_I.

$$E(\mu, P, R, K_I) = \frac{1}{2}\sum_{k=1}^{K} \hat{e}_{n+k}^2 + \frac{1}{2}\lambda[u_n - u_{n-1}]^2 \tag{5.43}$$

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

$$\begin{bmatrix} \mu 1_{new} \\ \mu 2_{new} \\ P_{new} \\ R_{new} \\ K_{I_{new}} \end{bmatrix} = \begin{bmatrix} \mu 1_{old} \\ \mu 2_{old} \\ P_{old} \\ R_{old} \\ K_{I_{old}} \end{bmatrix} + \left[J^T J + \eta I\right]^{-1} J^T e \tag{5.44}$$

where

$$\begin{aligned} 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 \end{aligned} \tag{5.45}$$

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

$$\hat{y}_{n+k} = f_n(\hat{c}_{n+k}) + g_n(\hat{c}_{n+k}) u_n \tag{5.46}$$

$$\hat{c}_{n+k} = [\underbrace{u_n, \cdots, u_n}_{\text{k}}, \underbrace{u_{n-1}, \cdots, u_{n+k-n_u}}_{n_u\text{-k}} \underbrace{\hat{y}_{n+k}, \cdots, \hat{y}_{n+1}}_{\text{k-1}} \underbrace{y_n, \cdots, y_{n+k-n_y}}_{n_y\text{+1-k}}] \tag{5.47}$$

where

$$f_n(\hat{c}_{n+k}) = \sum_{j \in SV} \alpha_j K(d_{n+k,j}) + b_f = \sum_{j \in SV} \alpha_j exp(\frac{-d_{n+k,j}}{2\sigma^2}) \tag{5.48}$$

$$g_n(\hat{c}_{n+j}) = \sum_{j \in SV} \theta_j K(d_{n+k,j}) + b_g = \sum_{j \in SV} \theta_j exp(\frac{-d_{n+k,j}}{2\sigma^2}) \tag{5.49}$$

$d_{n+k,j}$ is the Euclidean distance between the jth support vector x_j and the state vector at step $(n+k)$. The choice of the Kernel variable, σ, significantly affects how observable the data is in the feature space. Similar points in the input space spread out broadly in the feature space due to a small σ in the Kernel matrix emphasizing similarities among the input data. As a result, more support vectors are required to accurately reflect the dynamics of the system. On the other hand, different data points in the input space are projected closer together in the feature space by greater σ values. This may result in a decrease in the nonlinearity of the Kernel and a consequently less precise description of the dynamics of the system.

$$\begin{aligned} d_{n+k,j} &= \sqrt{(c_{n+k} - x_j)^T (c_{n+k} - x_j)} \\ &= \sqrt{D_{U_{n+k}} + D_{Y_{n+k}}} = \sqrt{A_{n+k}} \end{aligned} \tag{5.50}$$

where

$$\begin{aligned} D_{U_{n+k}} = &\sum_{i=1}^{min(k,n_u)} (u_n - x_{j,i})^2 \\ &+ \sum_{i=min(k,n_u)+1}^{n_u} (u_{n+min(k,n_u)-i} - x_{j,i})^2 - \delta(n_u - k) \end{aligned} \tag{5.51}$$

$$\begin{aligned} D_{Y_{n+k}} = &\sum_{i=1}^{min(k-1,n_y)} (\hat{y}_{n+k-i} - x_{j,n_u+i})^2 \\ + &\sum_{i=min(k-1,n_y)+1}^{n_y} (u_{n+min(k-1,n_y)+1-i} - x_{j,n_u+i})^2 - \delta(n_y + 1 - k) \end{aligned} \tag{5.52}$$

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

$$\frac{\partial \hat{y}_{n+k}}{\partial u_n} = \frac{\partial f_n(c_{n+k})}{\partial d_{n+k,j}} \frac{\partial d_{n+k,j}}{\partial u_n} + \frac{\partial g_n(c_{n+k})}{\partial d_{n+k,j}} \frac{\partial d_{n+k,j}}{\partial u_n} u_n + g_n(c_{n+k}) \tag{5.53}$$

where

$$\begin{aligned} \frac{\partial f_n(c_{n+k})}{\partial d_{n+k,j}} &= \frac{-1}{2\sigma^2} \sum_{j \in SV} \alpha_j exp(\frac{-d_{n+k,j}}{2\sigma^2}), \\ \frac{\partial g_n(c_{n+k})}{\partial d_{n+k,j}} &= \frac{-1}{2\sigma^2} \sum_{j \in SV} \theta_j exp(\frac{-d_{n+k,j}}{2\sigma^2}), \end{aligned} \tag{5.54}$$

and

$$\frac{\partial d_{n+k,j}}{\partial u_n} = \frac{\partial d_{n+k,j}}{\partial A_{n+k}} \left[\frac{\partial A_{n+k}}{\partial D_{U_{n+k}}} \frac{\partial D_{U_{n+k}}}{u_n} + \frac{\partial A_{n+k}}{\partial D_{Y_{n+k}}} \frac{\partial D_{Y_{n+k}}}{u_n} \right] \tag{5.55}$$

$$\frac{\partial d_{n+k,j}}{\partial A_{n+k}} = \frac{1}{2\sqrt{A_{n+k}}} = \frac{1}{2d_{n+k,j}}, \tag{5.56}$$

$$\frac{\partial A_{n+k}}{\partial D_{U_{n+k}}} = \frac{\partial A_{n+k}}{\partial D_{Y_{n+k}}} = 1, \tag{5.57}$$

$$\frac{\partial D_{U_{n+k}}}{\partial u_n} = \sum_{i=1}^{min(k,n_u)} 2(u_n - x_j, i), \tag{5.58}$$

$$\frac{\partial D_{Y_{n+k}}}{\partial u_n} = \sum_{i=1}^{min(k-1,n_y)} 2(\hat{y}_{n+k-i} - x_{j,n_u+i}) \frac{\partial \hat{y}_{n+k-i}}{\partial u_n} \delta(k-i) \tag{5.59}$$

According to the system's $LSSVR_{NARMA-L2}$ model, as illustrated in (5.53)-(5.59), the Jacobian matrix J_m is obtained. The performance of the proposed method relies on the parameters (μ_1, μ_2, P, R). However, to address observed steady-state error, an integrator has been introduced into the controller design. Additionally, to account for the non-optimal values of adaptive parameters during the transient state of the closed-loop system, a correction term has been introduced to compensate for any errors in the resulting control signal. This correction term is designed to accommodate and minimize errors in the resulting control signal caused by the non-optimal parameter values, thereby improving the system's performance during transients. Through the utilization of Taylor expansion, as outlined in references [123] and [124], the correction term is J_m derived as follows:

$$\delta u_n = -\left[J_m^T J_m\right]^{-1} J_m^T \hat{e} \tag{5.60}$$

By utilizing the Levenberg-Marquardt algorithm, the objective function presented in Equation (5.43) is optimized to fine-tune the parameters of the system. Moreover, the Levenberg-Marquardt algorithm is also employed to adapt the adjustable controller parameters. This dual application of the algorithm facilitates effective parameter tuning and controller adjustment, leading to improved performance of the system.

5.4 The Proposed Adaptive Inverse Optimal Controller Based on Least Squares Support Vector Regression for Nonlinear Non-Affine Systems

Below is a detailed description of the suggested control approach. In addition, Fig. (5.3) shows the control architecture of the entire system.

1. Set the controller and model parameters.
   -Determine initial model and controller μ_j, P, R, and K_I parameters.
   -$LSSVR_{NARX}$ model: β=b_β=0, $\sigma = 1, C = 1000$ $\varepsilon = 10^{-3}$

2. Utilizing the (u_{fn-1}, y_n) data pair, train the $LSSVR_{NARX}$ model
   -Set index to n
   -Compose x_{n-1}=$[y_{n-1}, \cdots, y_{n-n_y}, u_{n-2}, u_{n-n_u}]$
   - Predict $\hat{y}_n$ using the $LSSVR_{NARX}$ model
   - Compute $e_{m_n} = y_n - \hat{y}_n$, Given that $\mid e_{m_n} \mid > \varepsilon$, where $e_{m_n} = y_n - \hat{y}_n$, train the system model; otherwise proceed using the controller parameters and system model that were acquired in the preceding step.

3. Convert the $LSSVR_{NARX}$ model to the $LSSVR_{NARMA-L2}$ model. -Determine the parameters of $F_{n-1}(\alpha, b_f)$ and $G_{n-1}(\theta, b_g)$ using the $LSSVR_{NARX}$ parameters (β, b_β) displayed as (5.42)
   -Create $x_n = [u_{n-1}, \cdots, u_{n-n_u, y_n, \cdots, y_{n-n_y+1}}]$with these parameters.
   -Using $F_{n-1}(.)$ and $G_{n-1}(.)$, compute $\hat{f}_n^-$ and $\hat{g}_n^-$.

4. Use the inverse optimal controller with integrator (u_{c_n}) to calculate the control input that the controller produces. Determine the control signal (u_{c_n}) by utilizing the control law and submodel predictions found in (3.9).

5. Utilize the control signal u_{c_n} in order to calculate y_{n+1}. The $LSSVR_{NARX}$ model (u_{c_n}, y_{n+1}) training data pair is obtained for the following step.

6. Calculate the Jacobian.

-To create the Jacobian matrix in (5.45), apply u_{c_n} K times to the $LSSVR_{NARX}$ model.

7. Update controller parameters (μ_j, P, R, K_I) using (5.44), this is the first step in the controller learning process.

-Step n=n+1

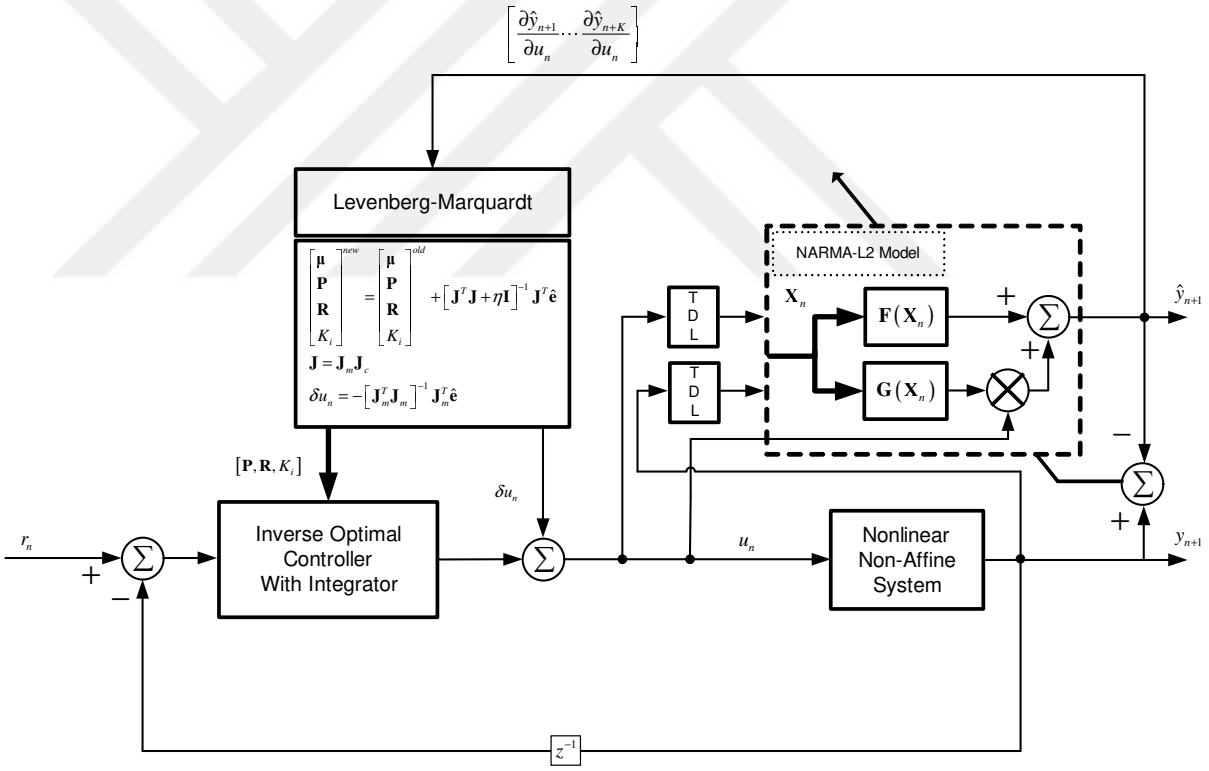

Figure 5.3 : Proposed Least Squares Support Vector Regression based Controller Scheme

5.5 Simulation Results For Least Squares Support Vector Regression Based Adaptive Inverse Optimal Controller For Nonlinear Non-Affine Systems

The control methodology proposed in this thesis has been evaluated through simulations conducted on two distinct nonlinear non-affine benchmark problems. The performance of the control systems has been assessed under various scenarios, such as nominal conditions, measurement noise, input disturbance, and parametric uncertainty cases.

Under the assumption that the system dynamics are unknown, the initial approach involved utilizing online $LSSVR_{NARX}$ to identify and detect these dynamics using input-output data pairs. Through the proposed method, the $LSSVR_{NARMA-L2}$ model has been derived as described by equations (5.35) to (5.42) and equation (5.44). For the $LSSVR_{NARX}$ and $LSSVR_{NARMA-L2}$ models, the input feature vectors have been determined as: $M_c = [u_{n-1}, \cdots, u_{n-n_u}, y_n, ..y_{n-n_y+1}]^T$ where $n_u = 3$ and $n_y = 3$ denote the number of previous inputs and outputs, respectively. The optimization of the inverse optimal controller parameters relies on a predictive objective function, as depicted in equation (5.43). To determine the optimal estimation horizon (K) and penalty term (λ), the grid search algorithm is employed. By systematically searching through different combinations of K and λ, the grid search algorithm helps to identify the values that result in the best performance for the inverse optimal controller. After employing the grid search algorithm, the optimal values of K and λ for system 1 were determined to be $K = 1$ and $\lambda = 0.1000$, while for system 2, the optimal values were found to be $K = 5$ and $\lambda = 0.2000$. These particular parameter settings were identified as the most effective for achieving enhanced performance for each respective system using the inverse optimal controller. The 3-D error surfaces are represented as a result of the grid search algorithm in Fig.(5.4) and Fig.(5.5) for system 1 and system 2, respectively.

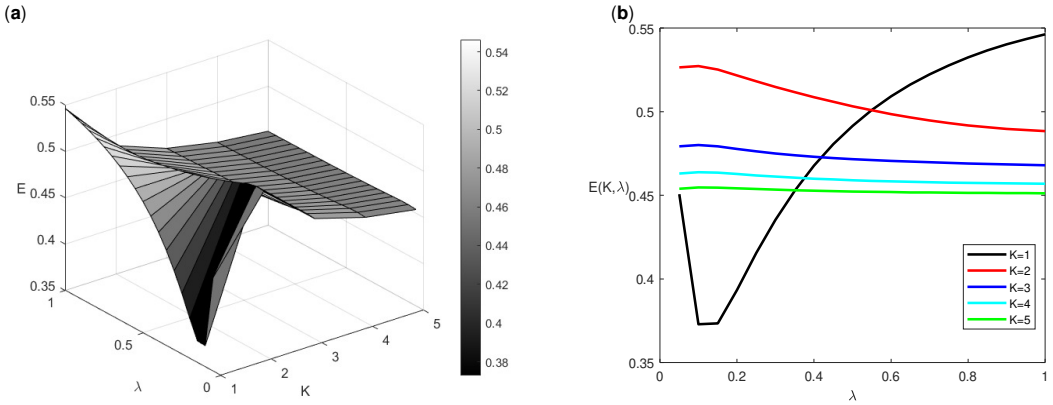

Figure 5.4 : 3-D Error Surface via Grid Search for Benchmark System I .

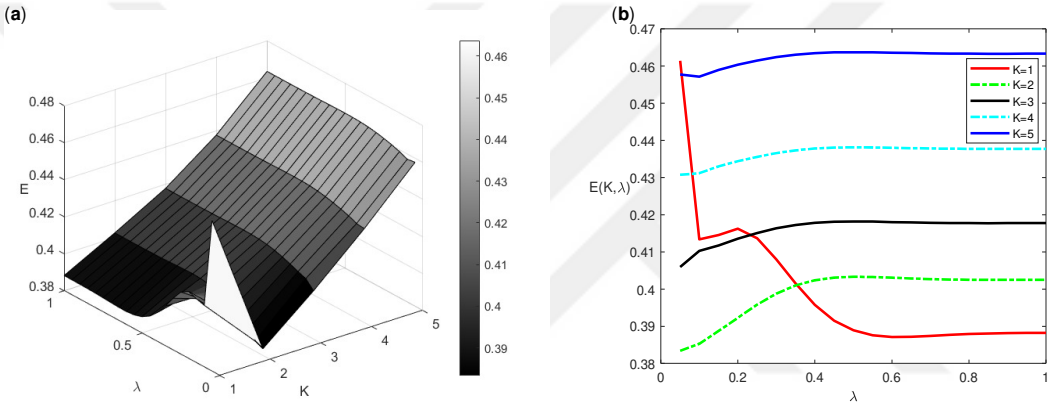

Figure 5.5 : 3-D Error Surface via Grid Search for Benchmark System II (CSTR) .

5.5.1 Benchmark system I

5.5.1.1 Nominal case

First, the following non-affine nonlinear system, which is characterized by:

$$x_{1_{n+1}} = 0.1x_{1_n} + \frac{2(u_n + x_{2_n})}{(1 + (u_{n+1} + x_{2_n})^2)} \tag{5.61}$$

$$x_{2_{n+1}} = 0.1x_{2_n} + u_n \frac{(2 + u_n^2)}{(1 + x_{1_n}^2 + x_{2_n}^2)} \tag{5.62}$$

$$y_{n+1} = x_{1_{n+1}} \tag{5.63}$$

was used to assess the effectiveness of the suggested strategy. Initial parameters are assigned as $P(0) = 0.000120$, $R(0) = 0.40$, $K_I(0) = 0.01$.

The tracking performance results for the nominal case are shown in Figs. (5.6) and (5.7). The states, control signal correction term, and control input law are shown in Fig. (5.6). P, Q, and K_I parameters, as well as μ_1 and μ_2, are given in Fig. (5.7). The trajectory tracking error, the Lyapunov function, and its derivative are depicted in Fig. (5.8). It is clearly seen that the trajectory tracking error is extremely small in magnitude. Additionally, the proven stability of the suggested controller for benchmark system I under nominal conditions is supported by the fact that the Lyapunov function is always positive and its derivative is always negative.

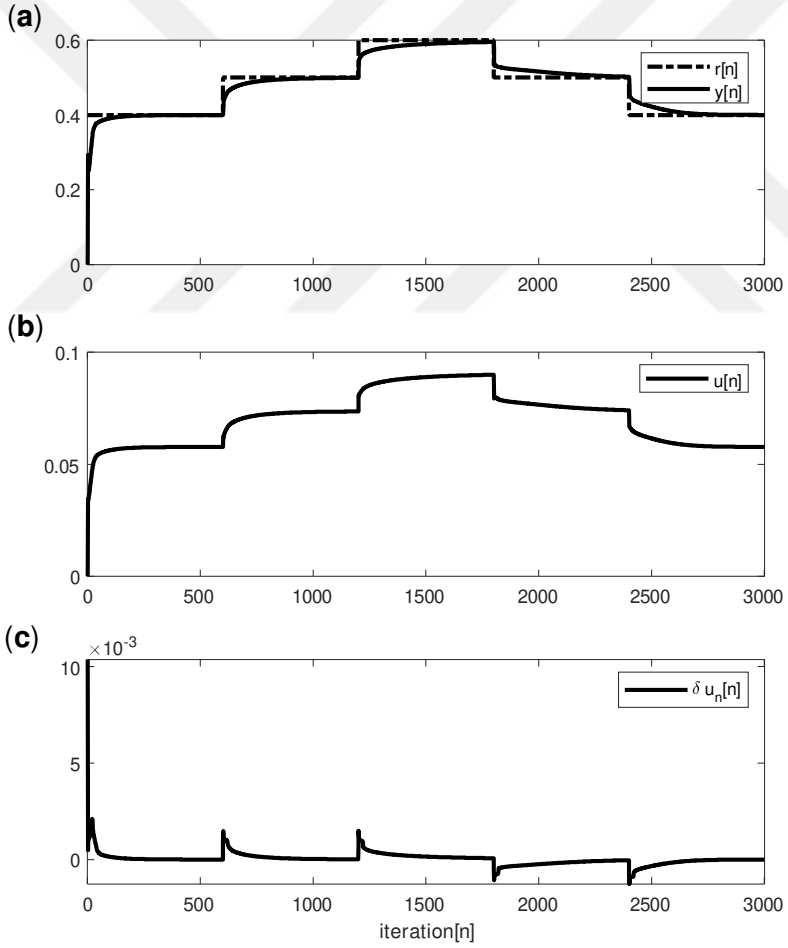

Figure 5.6 : Nominal case trajectory tracking performance results of benchmark system I

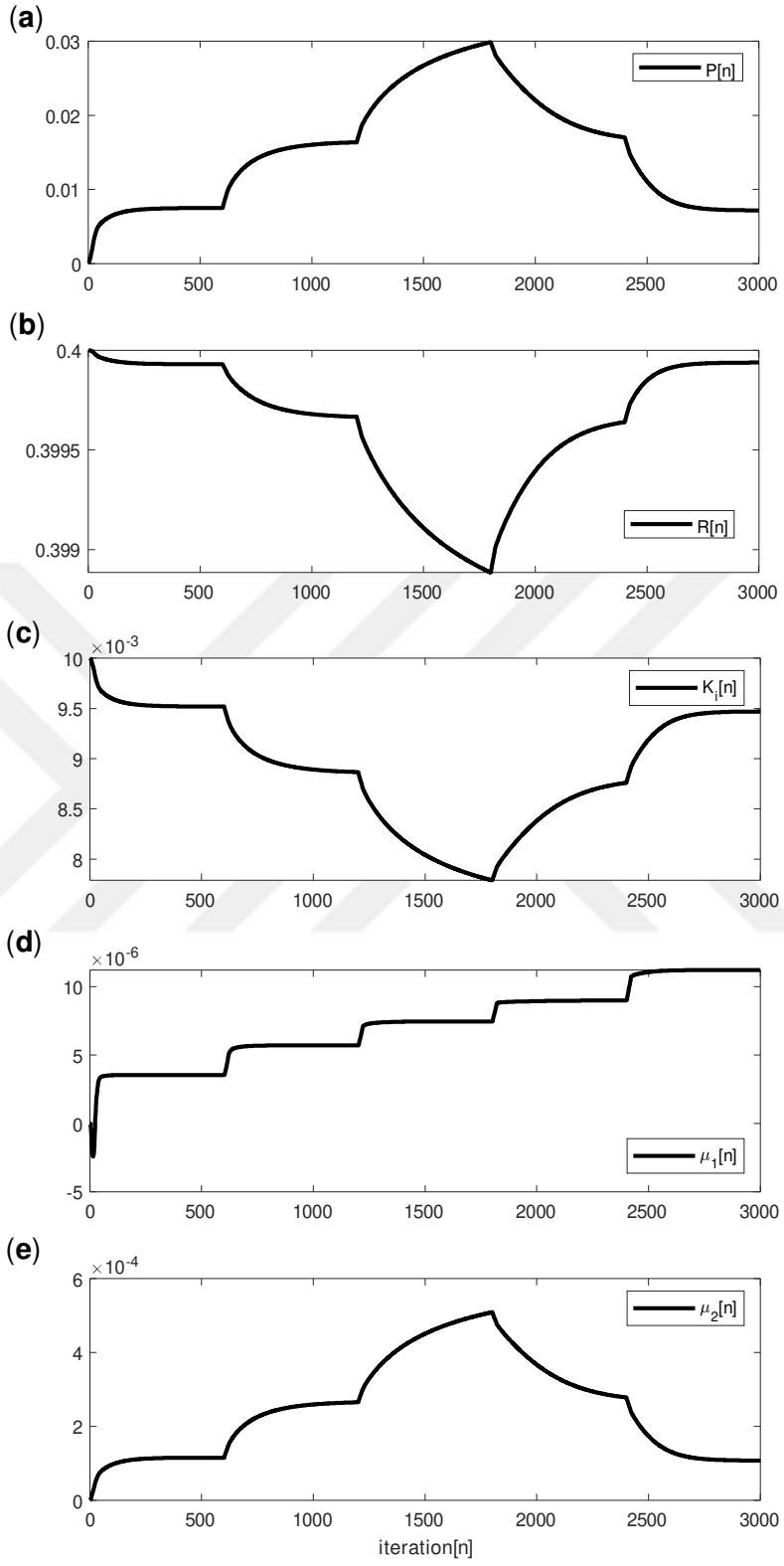

Figure 5.7 : Nominal case adaptive parameter results of benchmark system I

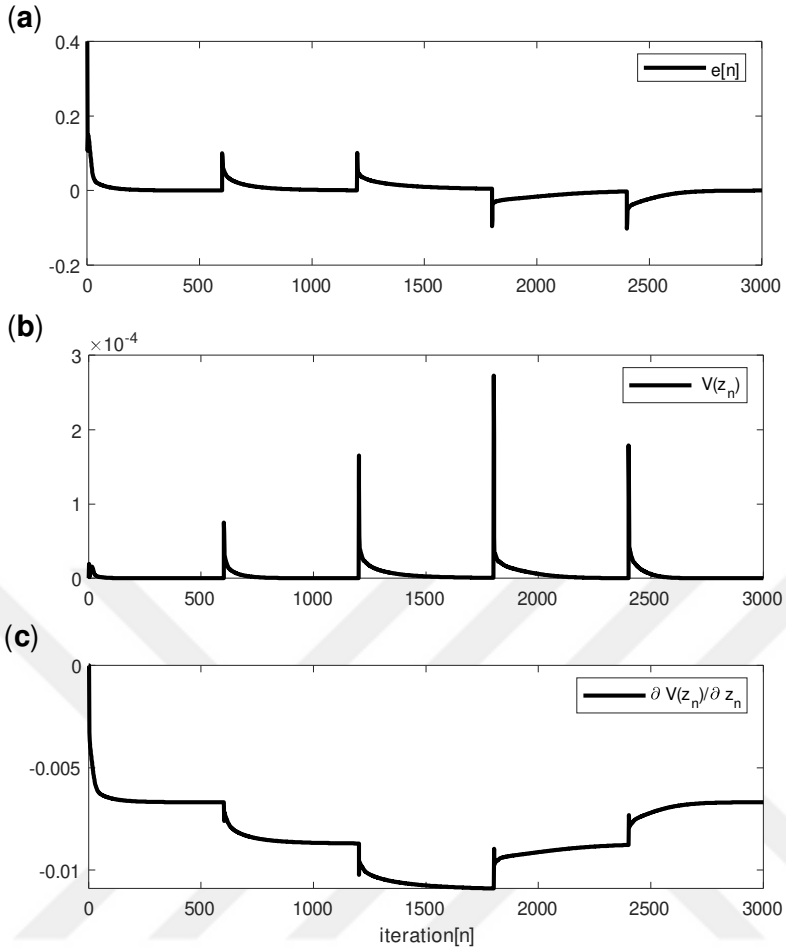

Figure 5.8 : Nominal case trajectory Tracking Error (**a**), Stability analysis (**b**), (**c**) of benchmark system I .

5.5.1.2 Measurement noise case

The output of the benchmark system I is subjected to 30 dB measurement noise in order to assess the trajectory tracking performance of the proposed control technique. (5.9) The computed control input, control signal correction term, and system output are displayed in Fig(5.9). The parameters of the inverse optimal controller with integrator and $LSSVR_{NARMA-L2}$ decomposition parameters are expressed in Fig. (5.10). It is clear that the proposed controller is competent to handle measurement noise and can self-tune parameters to execute control.

Fig. (5.11) displays the Lyapunov function, its derivative, and the trajectory tracking error. It is clear that the suggested controller keeps benchmark system I stable even in the presence of measurement noise.

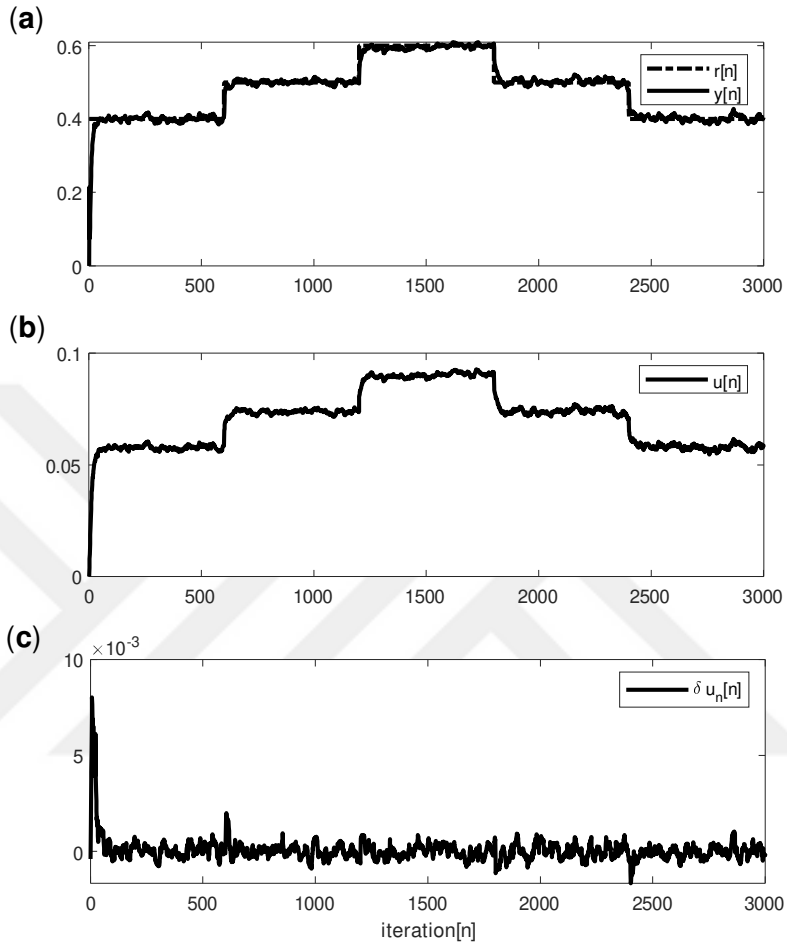

Figure 5.9 : Measurement noise case trajectory tracking performance results of benchmark system I .

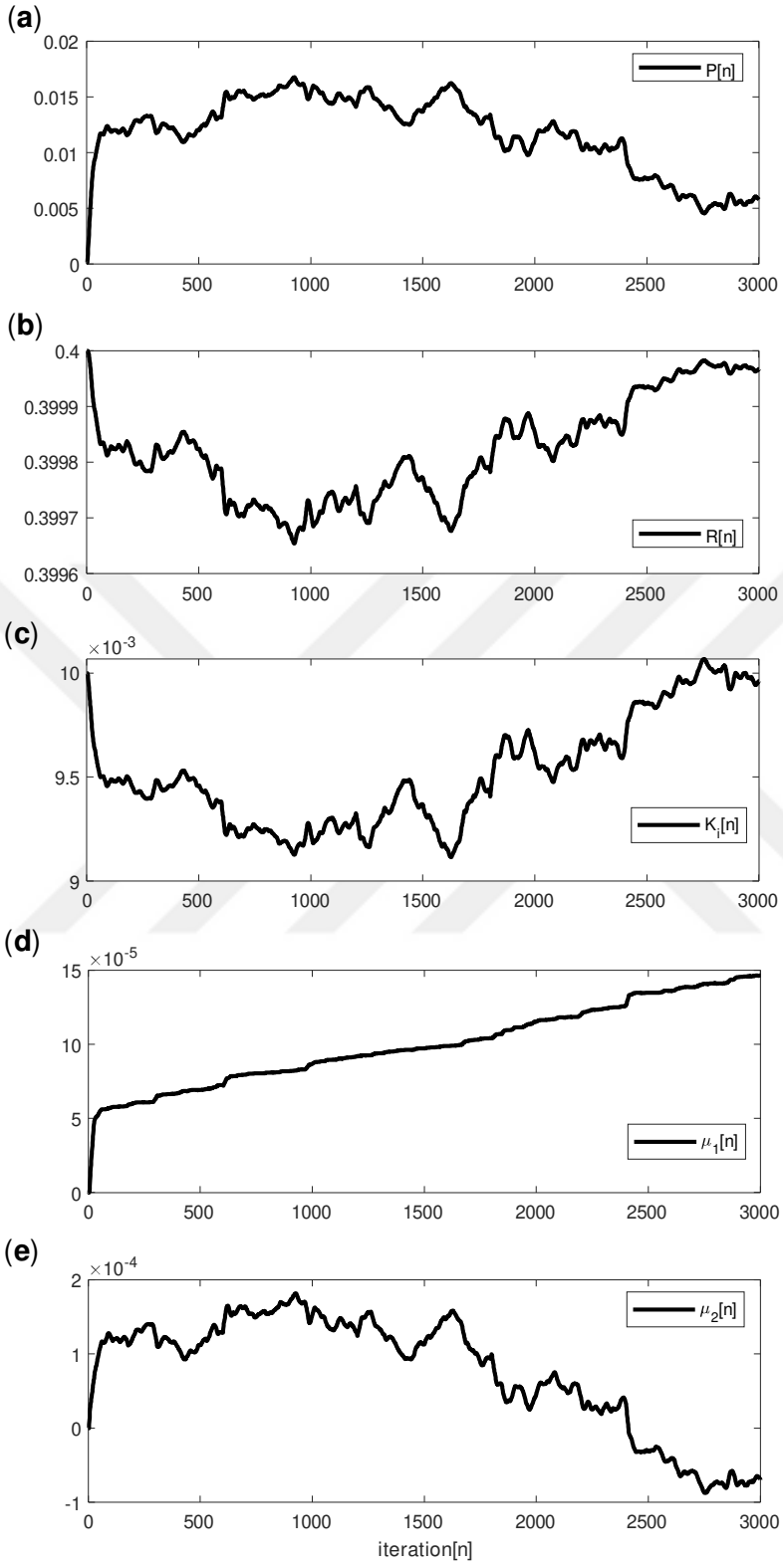

Figure 5.10 : Measurement noise case adaptive parameter results of benchmark system I .

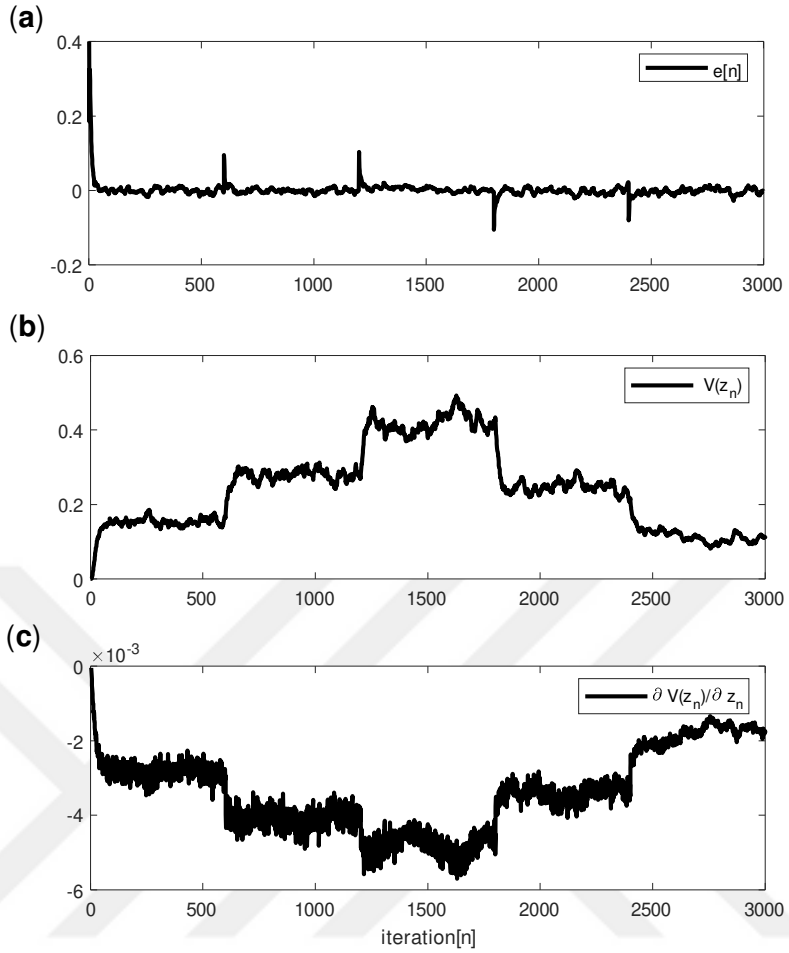

Figure 5.11 : Measurement noise case trajectory tracking error (**a**), stability analysis (**b**), (**c**)of benchmark system I.

5.5.1.3 Input disturbance case

The case where an input disturbance of $u_{d_n} = 0.01\sin(\frac{2\pi}{12.5}n)$ is added is analysed via simulations. Fig. (5.12) displays the tracking performance. Fig. (5.13) shows the adaptive response of the inverse optimal controller and decomposition parameters. The statistics unequivocally demonstrate that the suggested control mechanism is capable of both successfully delivering good control and handling input disturbance. The simulation results for the trajectory tracking error and stability obtained by the suggested method for the benchmark system I under input disturbance are shown in Fig. (5.14).

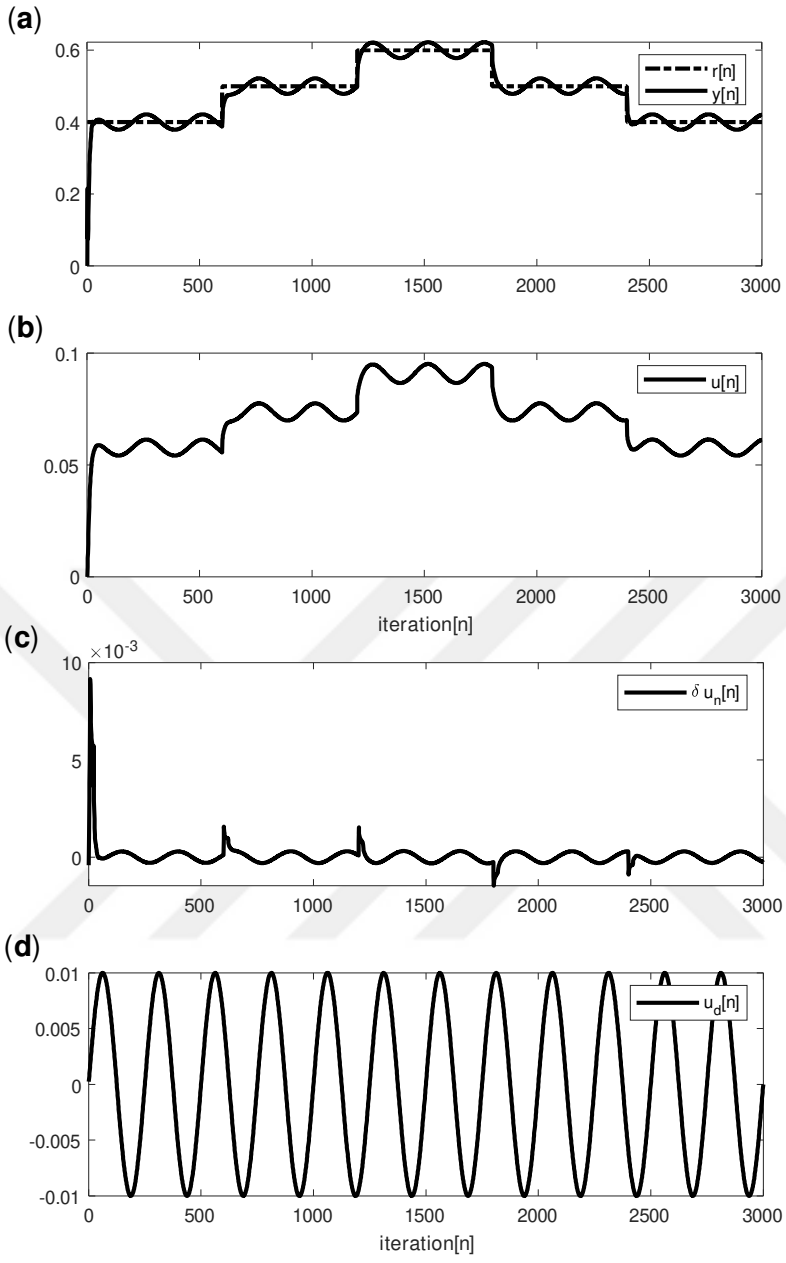

Figure 5.12 : Input disturbance case trajectory tracking performance results of benchmark system I .

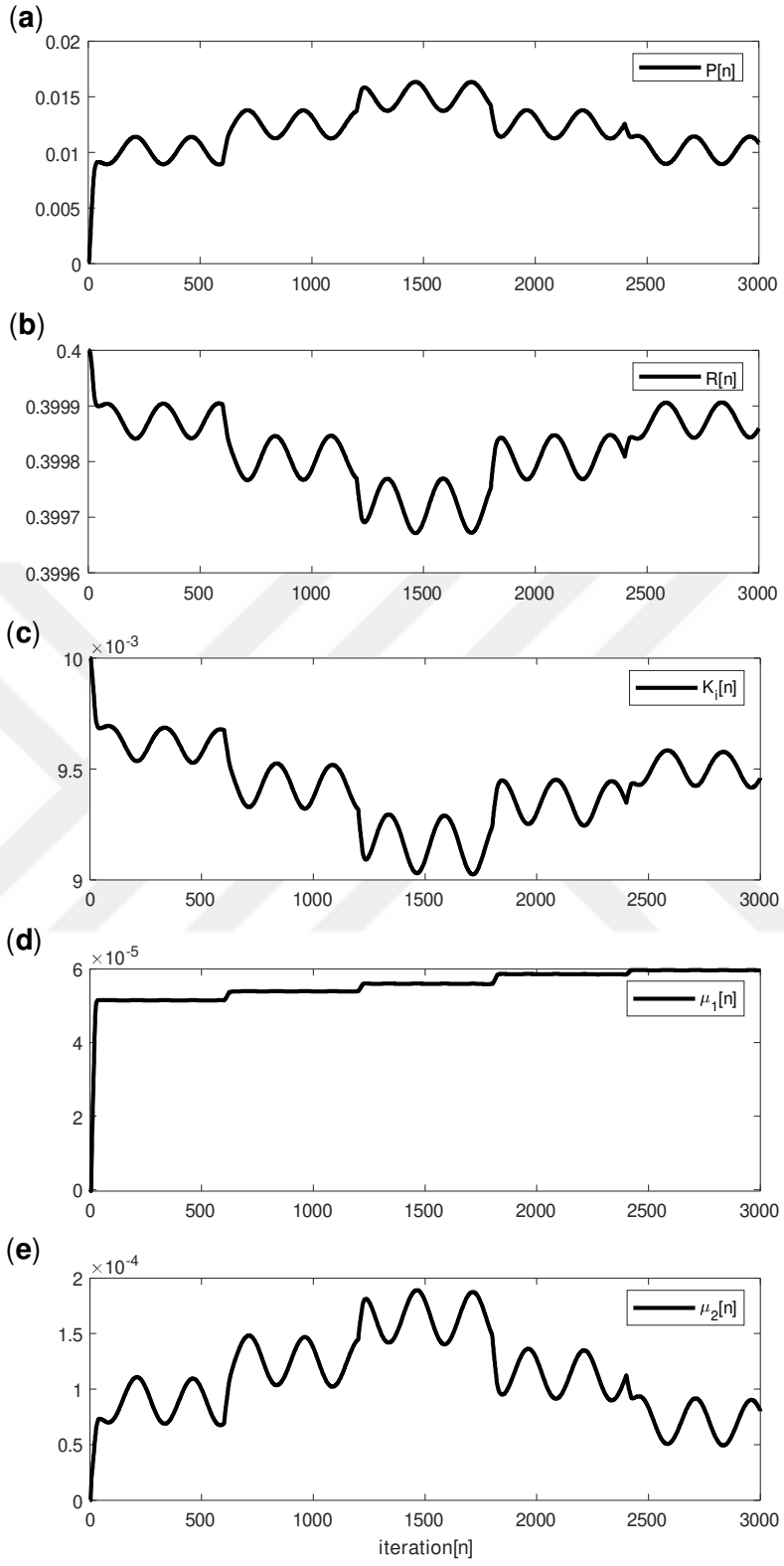

Figure 5.13 : Input disturbance case adaptive parameter results of benchmark system I

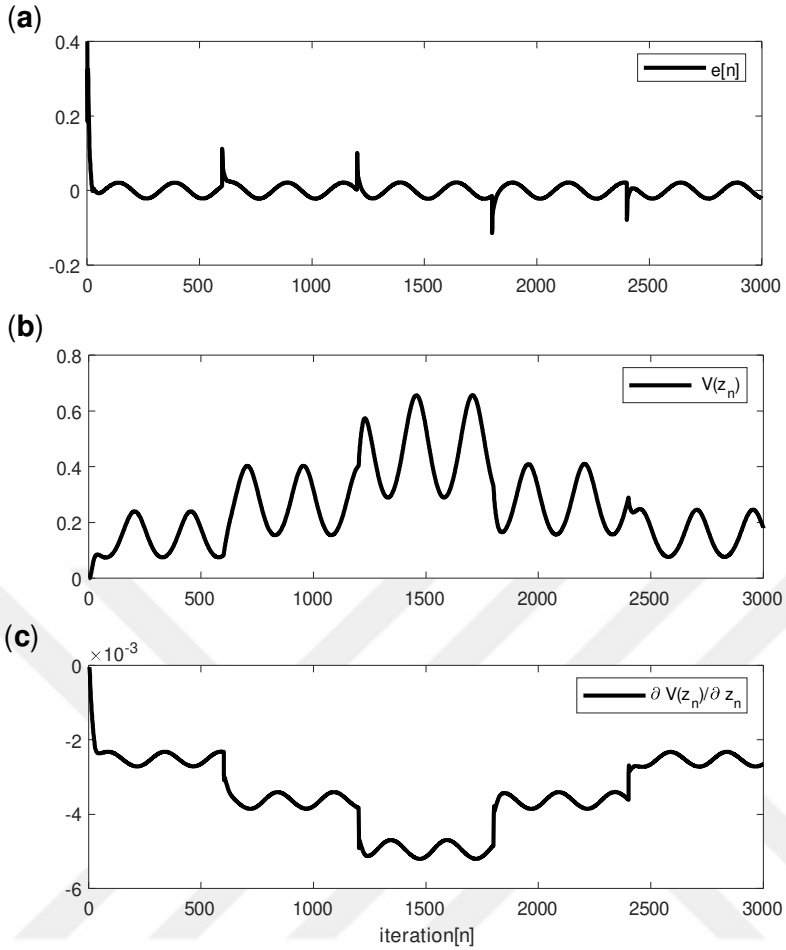

Figure 5.14 : Input disturbance case trajectory Tracking Error (**a**), stability analysis (**b**), (**c**) of benchmark system I.

5.5.1.4 Parametric uncertainty case

Uncertainty has been added by generating $p_n x_{1_n}$, where $p_n = 0.1(1+0.1sin(2\pi\frac{1}{25}n))$ is the uncertainty parameter for the coefficient of the term "$0.1x_{1_n}$" in (5.61). Fig. (5.15) shows the tracking control performance and the adjustment of the uncertain parameter. Fig. (5.16) displays the parameters of the inverse optimal controller with integrator and $LSSVR_{NARMA-L2}$ decomposition. The figures demonstrate that the controller can successfully achieve control despite the parametric uncertainty. In order to achieve effective control, the controller parameters are adaptively tuned.

Fig.(5.17) illustrates the stability and trajectory tracking error of benchmark system I with the addition of parametric uncertainty.

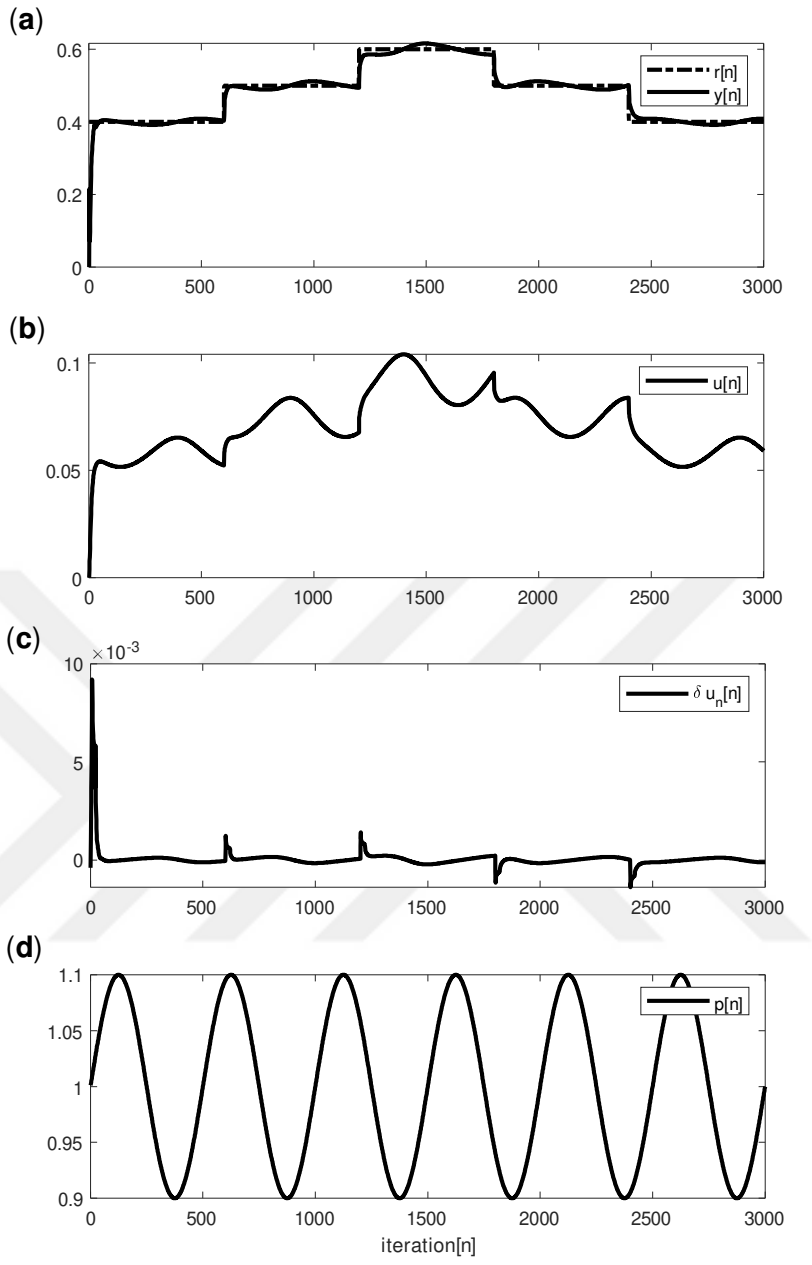

Figure 5.15 : Parametric uncertainty case trajectory tracking performance results of benchmark system I

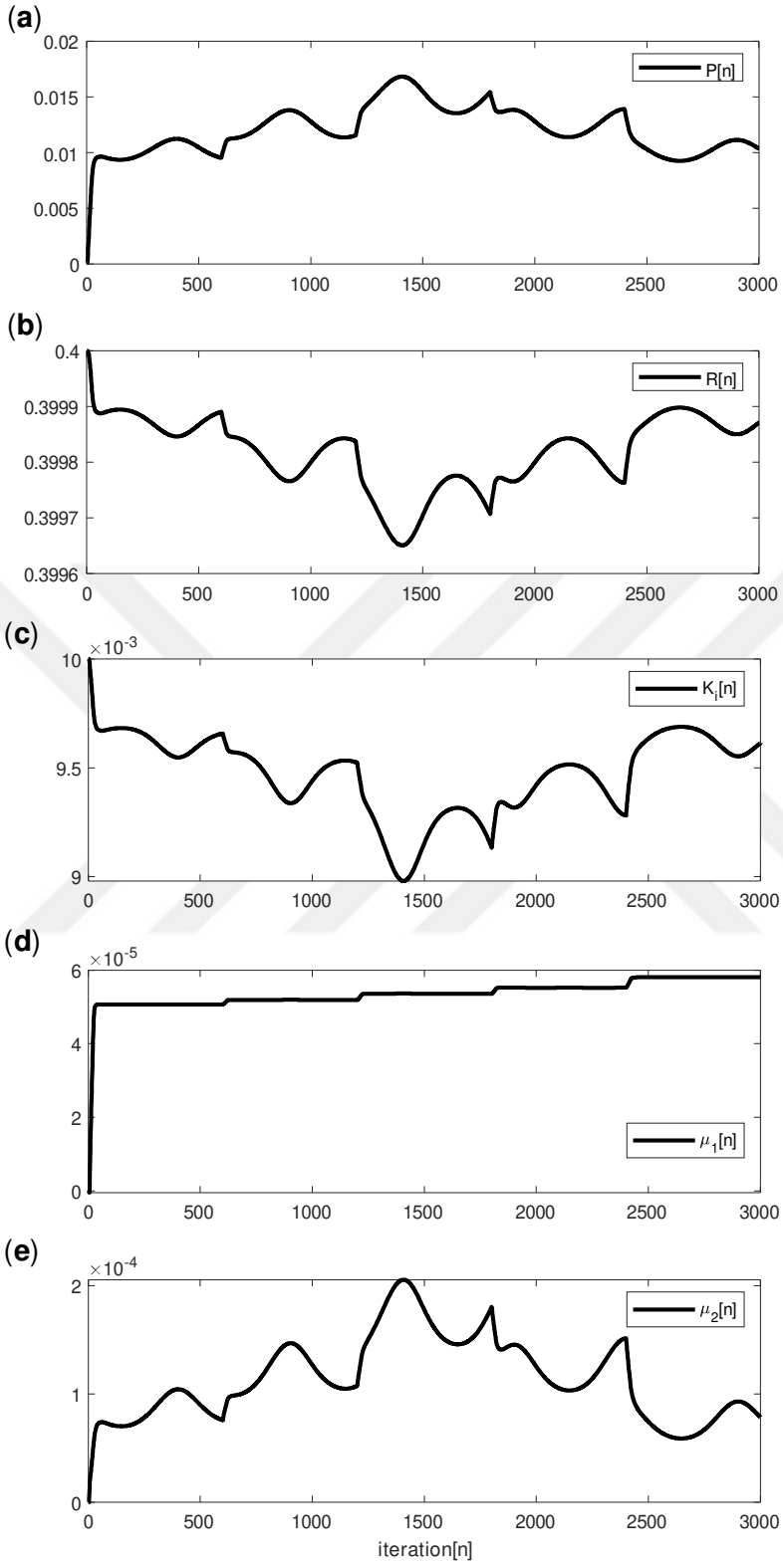

Figure 5.16 : Parametric uncertainty case adaptive parameter results of benchmark system I

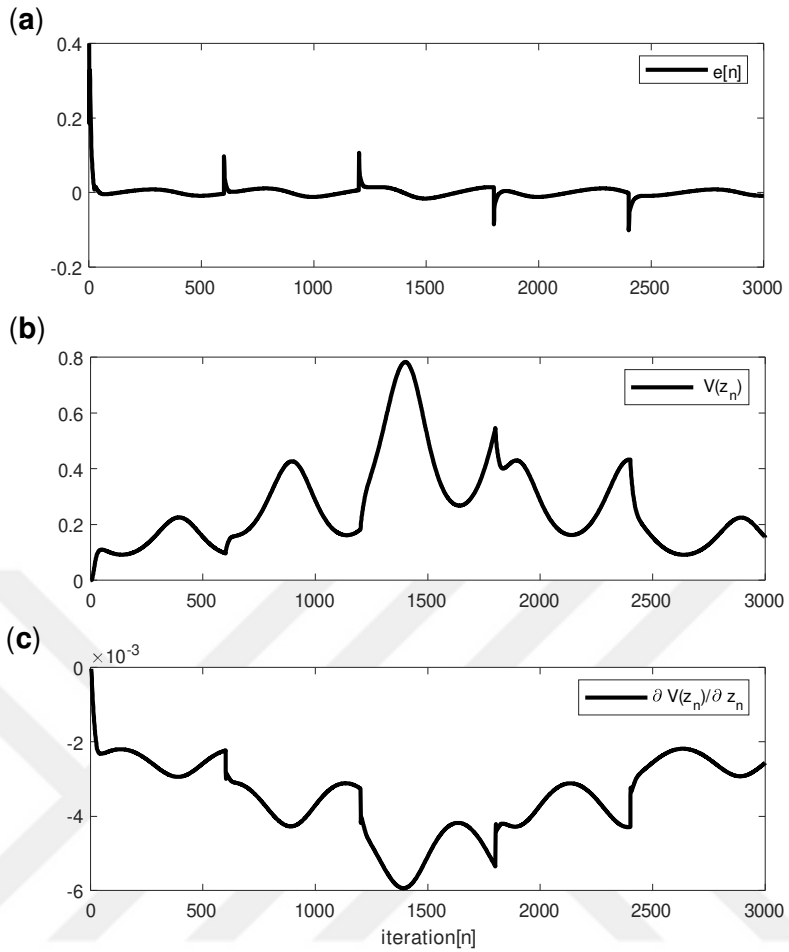

Figure 5.17 : Parametric uncertainty case trajectory tracking error(**a**), stability analysis (**b**), (**c**) of benchmark system I .

5.5.2 Benchmark system II

A third-order continuously stirred tank reactor (CSTR) system has been used to assess the performance of the suggested approach. The manufacture of continuous hydrogen is usually carried out in continuously stirred tank reactors (CSTRs) [125]. In addition to chemicals entering the reactor to undergo a reaction, there is also an exit stream that removes chemicals from the reactor to ensure the volume remains unchanged. Hence, a primary benefit of CSTRs is their capability to ensure uninterrupted production, they offer continuous output without constantly filling and emptying the reactor. [126]. Due to the inherent nonlinearity and non-affinity of the dynamical equations governing the CSTR system, controlling it poses a challenging

problem. This very challenge has consistently led to the CSTR system being adopted as a benchmark problem, serving to assess the capabilities of newly proposed control methodologies [124, 127]. The differential equations of the system are given as [124, 126, 127]:

$$\dot{x}_1(t) = 1 - x_1(t) - Da_1 x_1(t) + Da_2 x_2^2(t) \tag{5.64}$$

$$\dot{x}_2(t) = -x_2(t) + Da_1 x_1(t) - Da_2 x_2^2(t) - Da_3 d_2(t) x_2^2(t) + u(t) \tag{5.65}$$

$$\dot{x}_3(t) = -x_3(t) + Da_3 d_2(t) x_2^2(t) \tag{5.66}$$

where $d_{2_{\text{nominal}}}(t) = 1$, $Da_1 = 3$, $Da_2 = 0.5$, and $Da_3 = 1$. $u(t)$ is the system's input, and $x_3(t)$ is the process's output. A model exhibiting affine aspects of nonlinearity characterizes the CSTR system. In order to evaluate the performance of our suggested method for nonlinear non-affine systems, we changed the system input signal ($u(t)$) in (5.64,5.65,5.66) as $sin(\frac{2\pi}{4}u^2(t))$. Consequently, a non-affine system has been developed to evaluate the effectiveness of the newly implemented control framework.

The controller's initial parameter settings are $P(0) = 0.58$, $R(0) = 0.04$, and $K_I(0) = 0.01$Ṫhe simulation results for the nominal case, as well as the scenarios in which the system was subjected to measurement noise, input disturbance, and parametric uncertainty, are displayed below. For every case, the trajectory tracking error, computed control input, control signal correction term, inverse controller and integrator parameters, $LSSVR_{NARMA-L2}$ decomposition parameters, and tracking results for the output are displayed. In addition, the Lyapunov function and its derivative are provided to show that the suggested method is stable.

5.5.2.1 Nominal case

The trajectory tracking results for the nominal scenario, which includes no noise, disturbance, or parametric uncertainty, are shown in Figs. (5.18) and (5.19). The findings show that the suggested approach offers good tracking performance, a fast response time, and no steady-state inaccuracy. The figures also show how the adaptive parameters were adjusted to achieve optimal tracking performance with the

least amount of inaccuracy. The trajectory tracking error is minimized, as shown in Fig. (5.20), and the Lyapunov function is consistently positive and its derivative is consistently negative, indicating that the suggested strategy ensures the stability of the CSTR system.

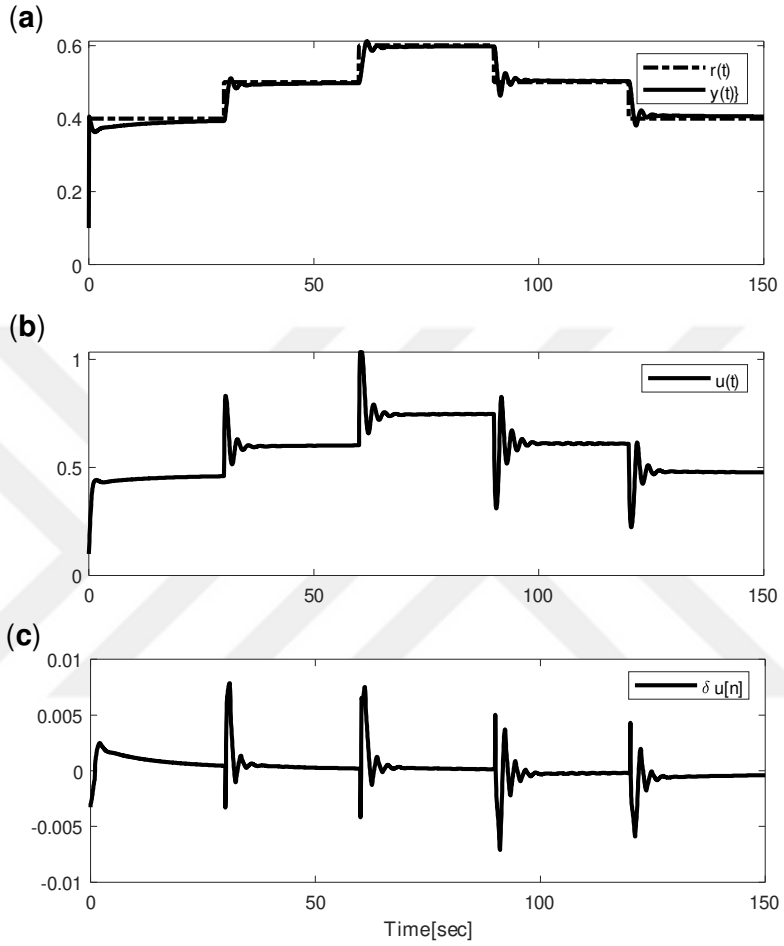

Figure 5.18 : Nominal case trajectory tracking performance results of benchmark system II (CSTR).

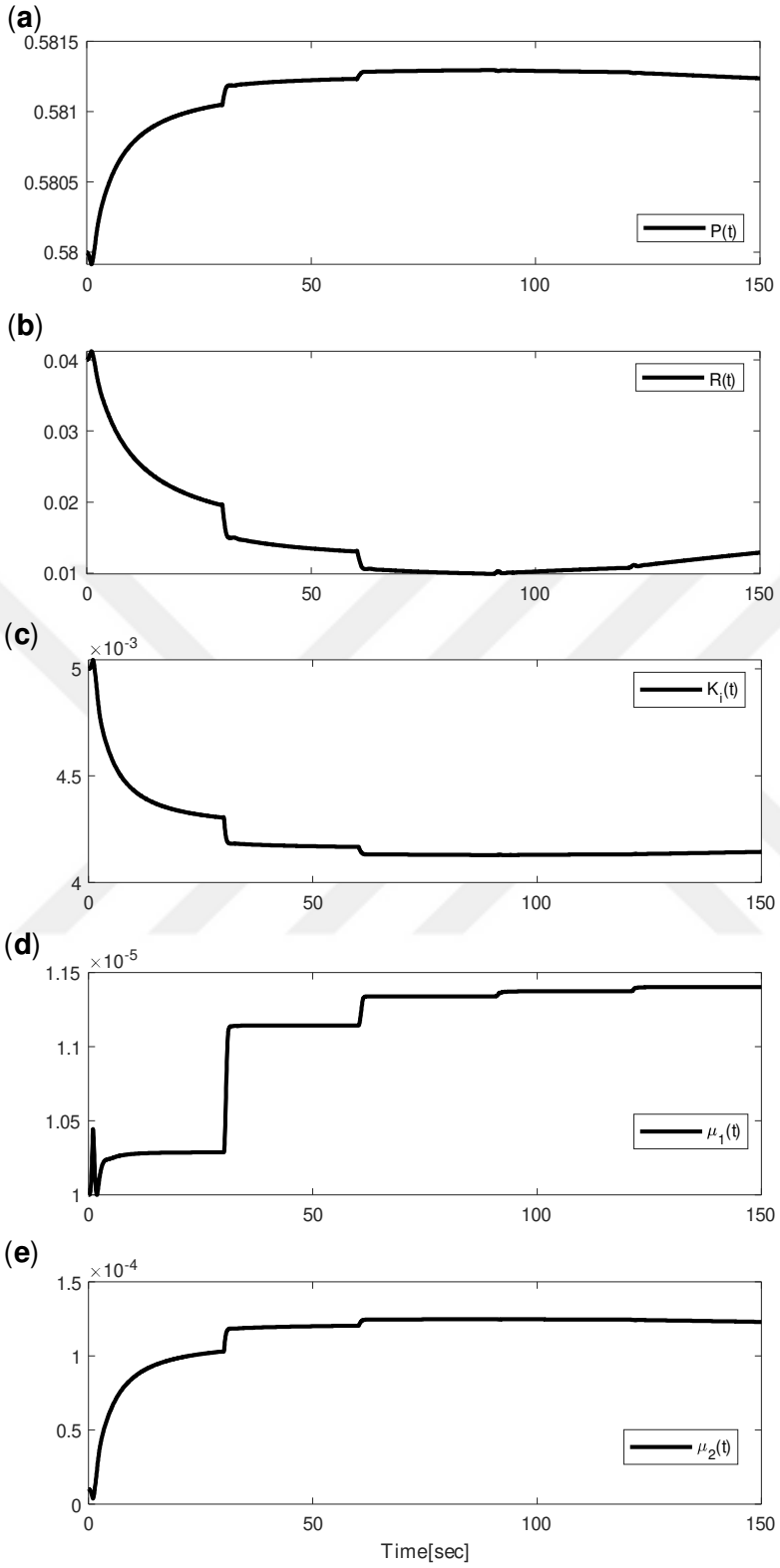

Figure 5.19 : Nominal case adaptive parameters of benchmark system II (CSTR).

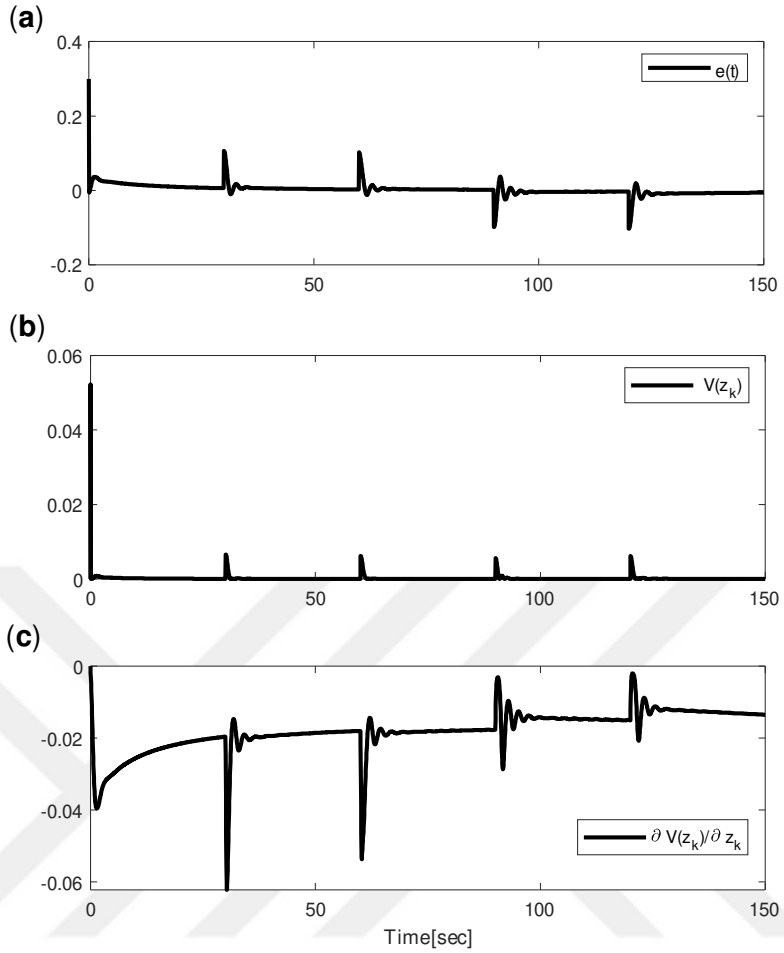

Figure 5.20 : Nominal case trajectory tracking error (**a**), stability analysis (**b**), (**c**), of benchmark system II(**b**), (**c**) (CSTR).

5.5.2.2 Measurement noise case

The benchmark system II (CSTR) output was subjected to 30 dB measurement noise in order to assess the performance of the suggested technique. The tracking performance of the suggested approach is shown in Fig (5.21), and the parameters of the inverse optimal controller with integrator and the parameters acquired for the $LSSVR_{NARMA-L2}$ system identification for the case with measurement noise are shown in Fig (5.22). The results of the simulations confirm that the suggested approach is effective in producing acceptable results even when system measurement noise is present.

The Lyapunov function and its derivative are displayed in Fig. (5.23), which establishes system stability and supports the objective of minimizing trajectory tracking error for the CSTR system in the presence of measurement noise.

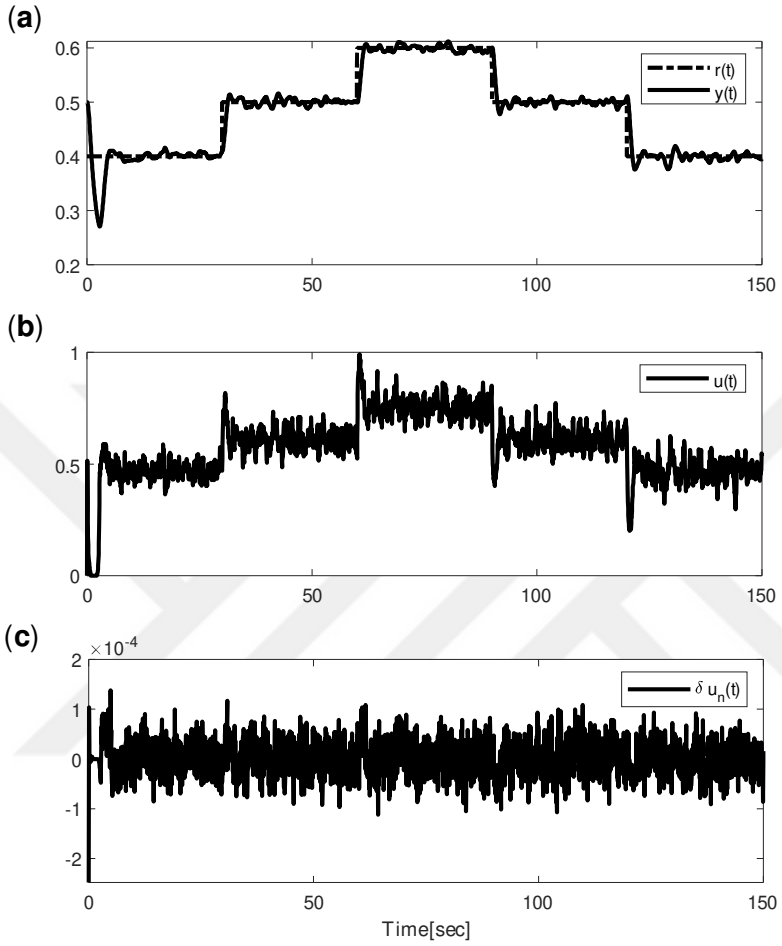

Figure 5.21 : Measurement noise case trajectory tracking performance results of benchmark system II (CSTR).

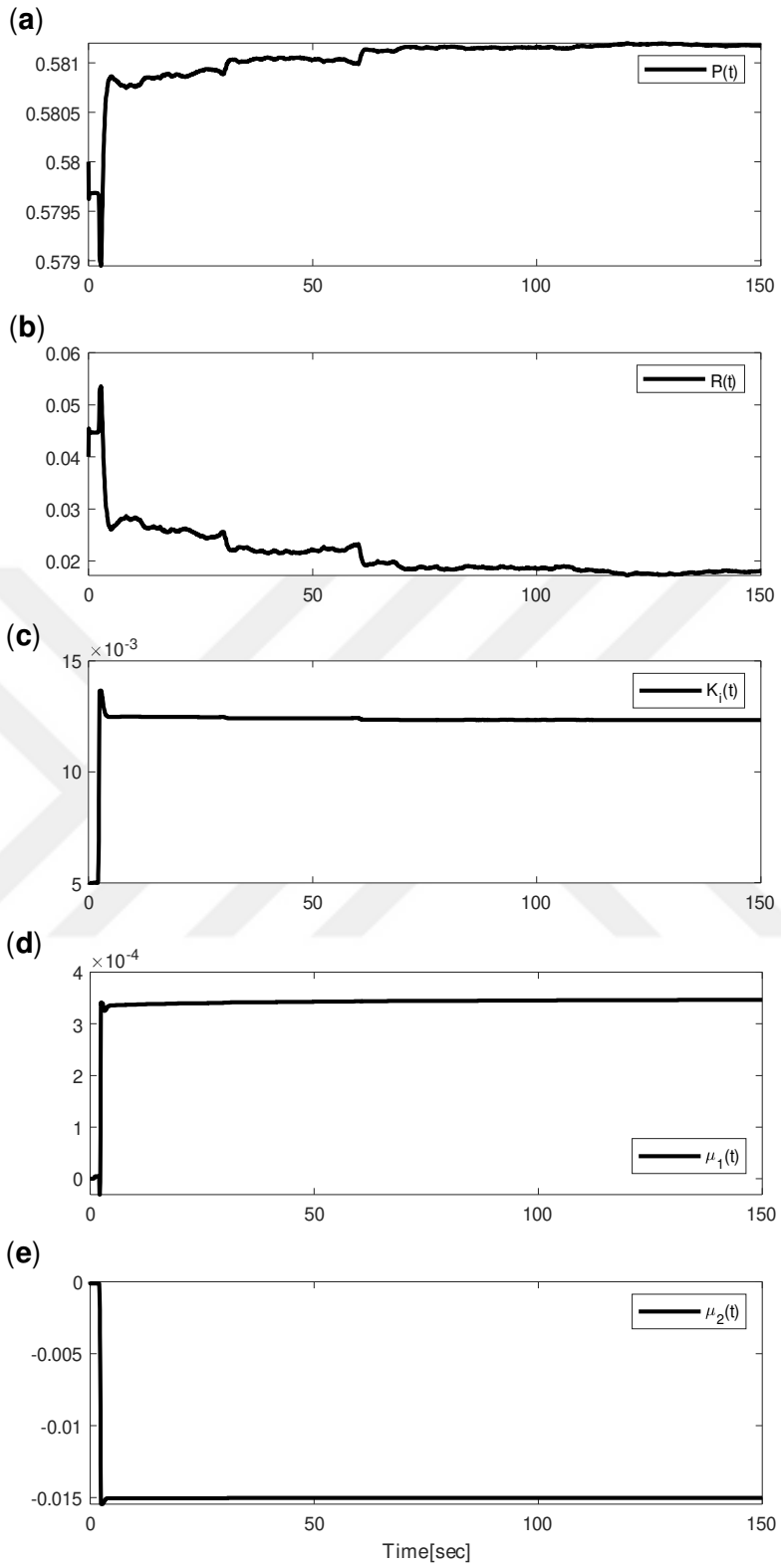

Figure 5.22 : Measurement noise case adaptive parameter results of benchmark system II (CSTR).

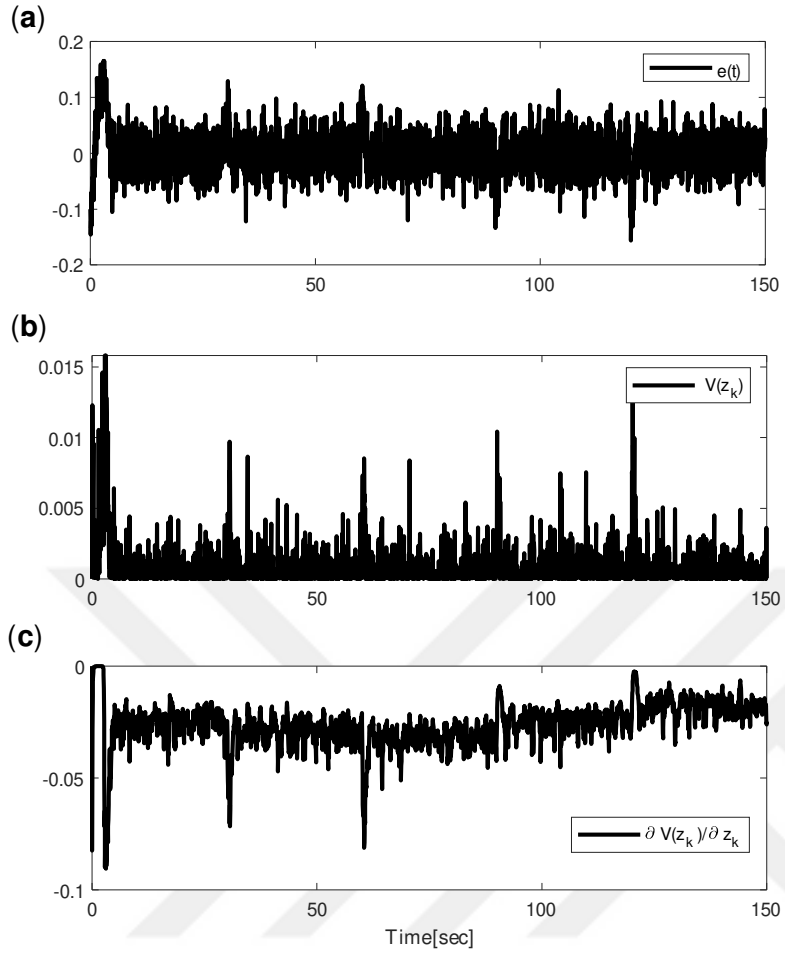

Figure 5.23 : Measurement noise case trajectory tracking performance results of benchmark system II (CSTR).

5.5.2.3 Input disturbance case

The performance of the proposed approach under input disturbance for the CSTR system is evaluated using an input disturbance of $d(t) = 0.01\sin(\frac{2\pi}{12.5}t)$. Fig. (5.24) illustrates the tracking performance success for the proposed approach under disturbance. The fine-tuning of the inverse optimal controller with integrator and $LSSVR_{NARMA-L2}$ system model decomposition parameters is shown in Fig. (5.25). The stability and tracking error findings for the scenario where an input disturbance is added to the CSTR system are shown in Fig. (5.26).

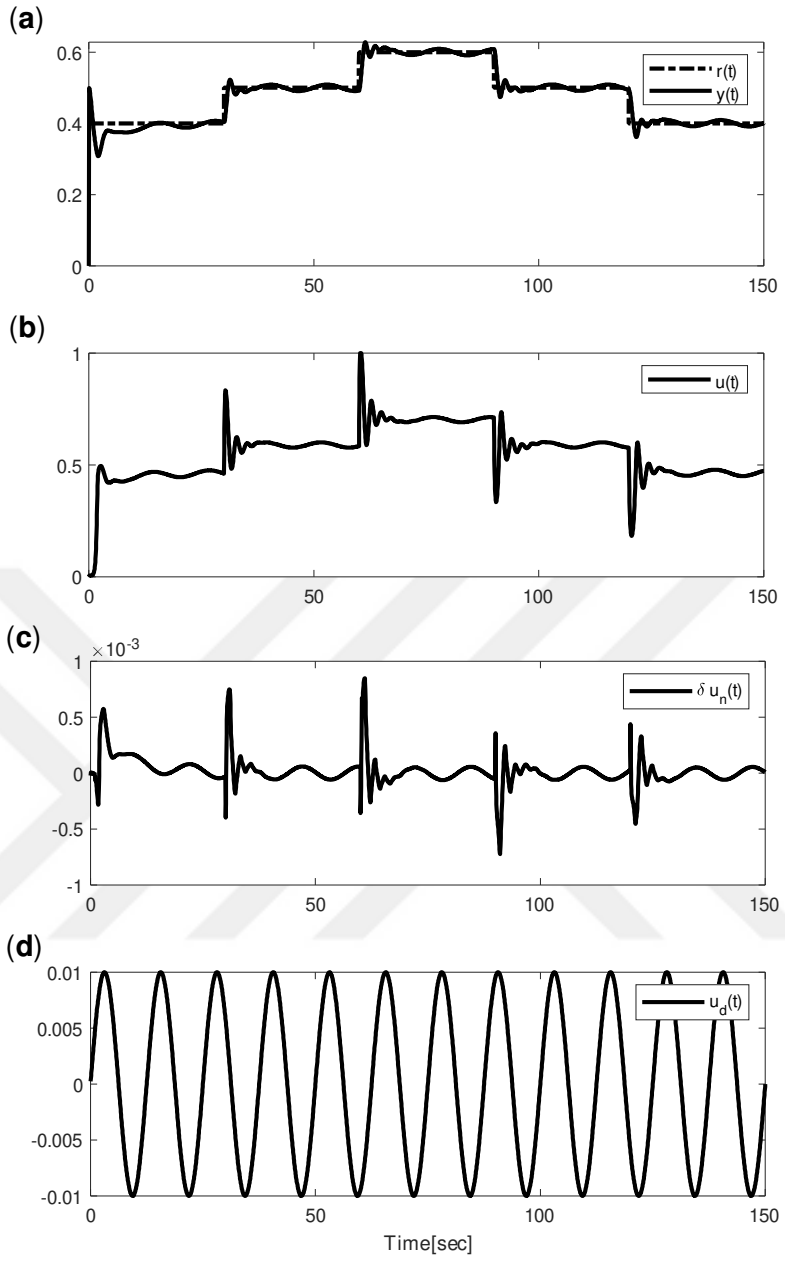

Figure 5.24 : Input disturbance case trajectory tracking performance results of benchmark system II (CSTR).

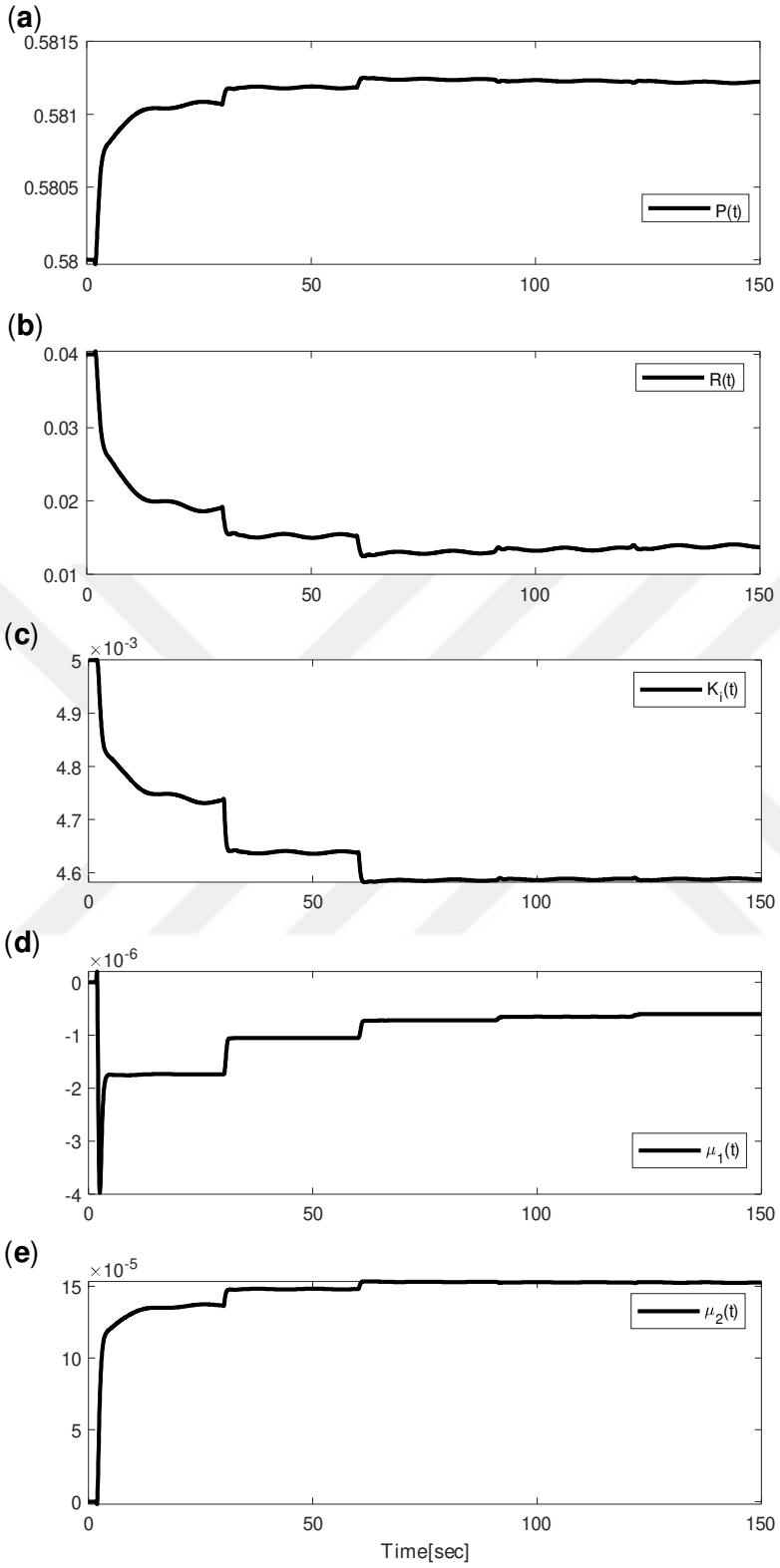

Figure 5.25 : Input disturbance case adaptive parameter results of Benchmark system II (CSTR).

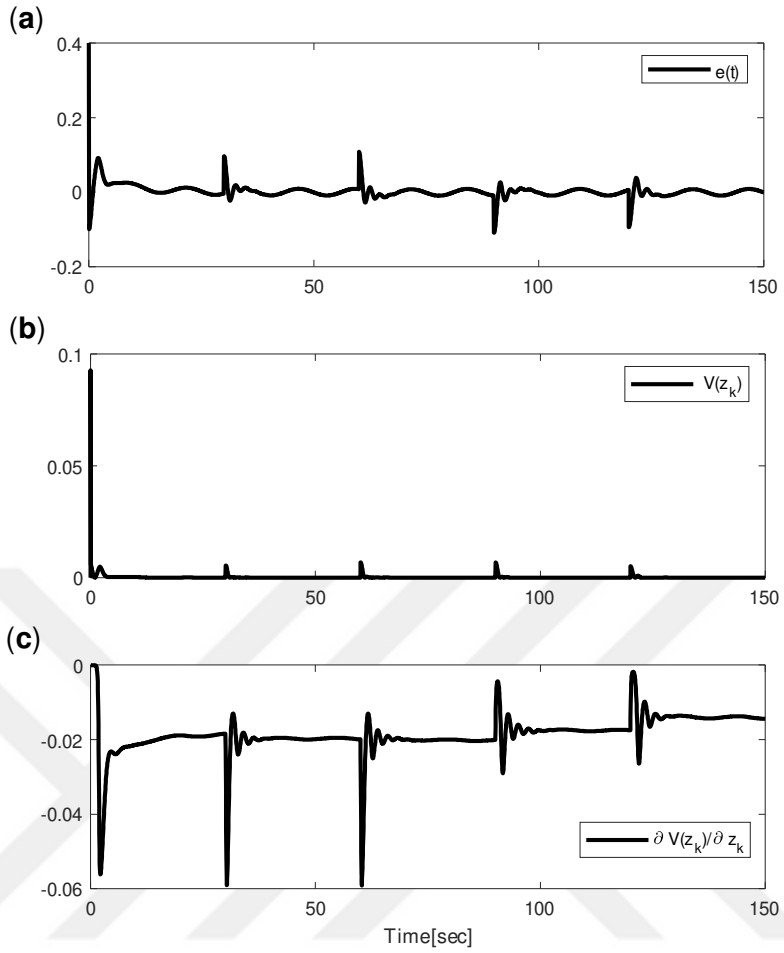

Figure 5.26 : Input disturbance case trajectory tracking error (**a**), stability analysis (**b**), (**c**) of benchmark System II (CSTR).

5.5.2.4 Parametric uncertainty case

The performance of the proposed control algorithm in the presence of parametric uncertainty in the system has also been examined. In this context, it is assumed that the parameter $d_2(t)$, which changes as $d_2(t) = (1 + 0.1\sin 0.08\pi t)$, is subject to a sinusoidal uncertainty. The results for the CSTR system show how highly successful the suggested approach is in reducing and neutralizing the effects of parametric uncertainties. Fig. (5.27) provides an illustration of tracking performance. Fig. (5.28) shows how the parameters of the inverse optimal controller with the integrator and $LSSVR_{NARMA-L2}$'s system identification parameters change.

The trajectory tracking error is illustrated in Fig. (5.29). In order to demonstrate the stability of the suggested control strategy when parametric uncertainty is added to the CSTR system, the figure additionally presents the Lyapunov function and its derivative.

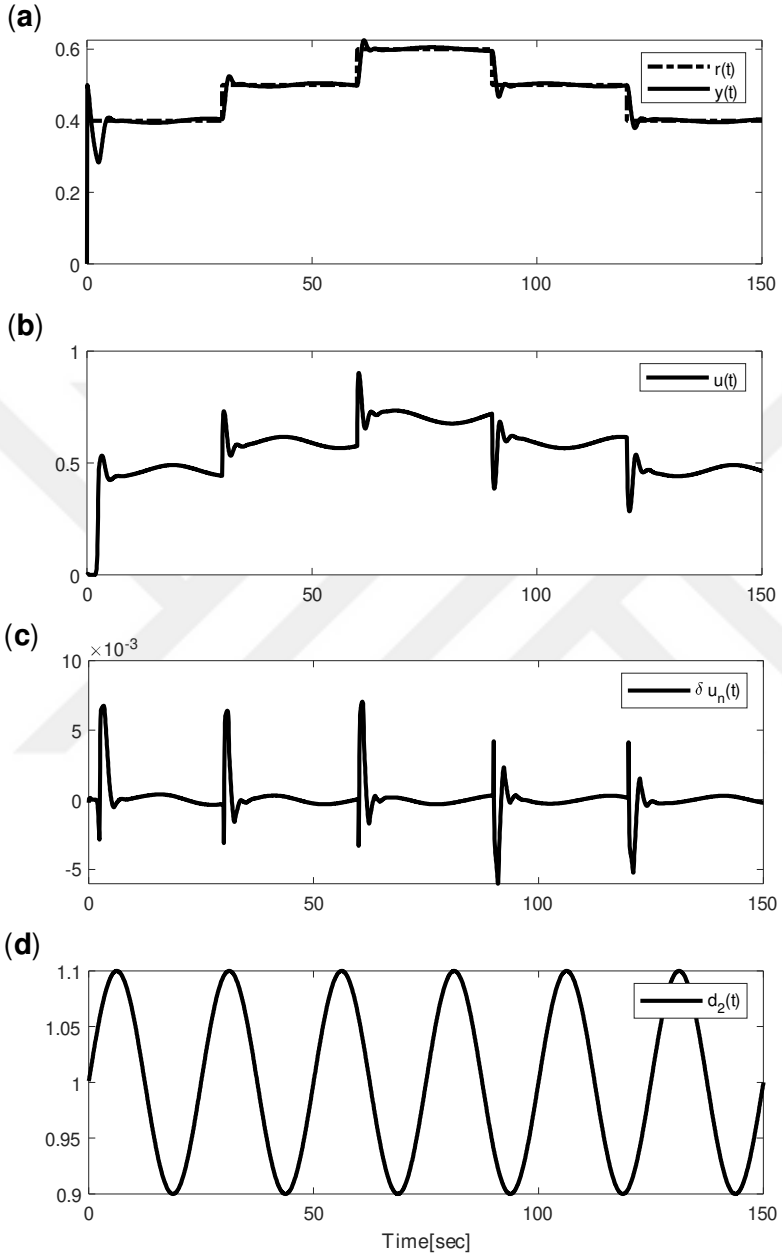

Figure 5.27 : Parametric uncertainty case trajectory tracking performance results of benchmark system II (CSTR).

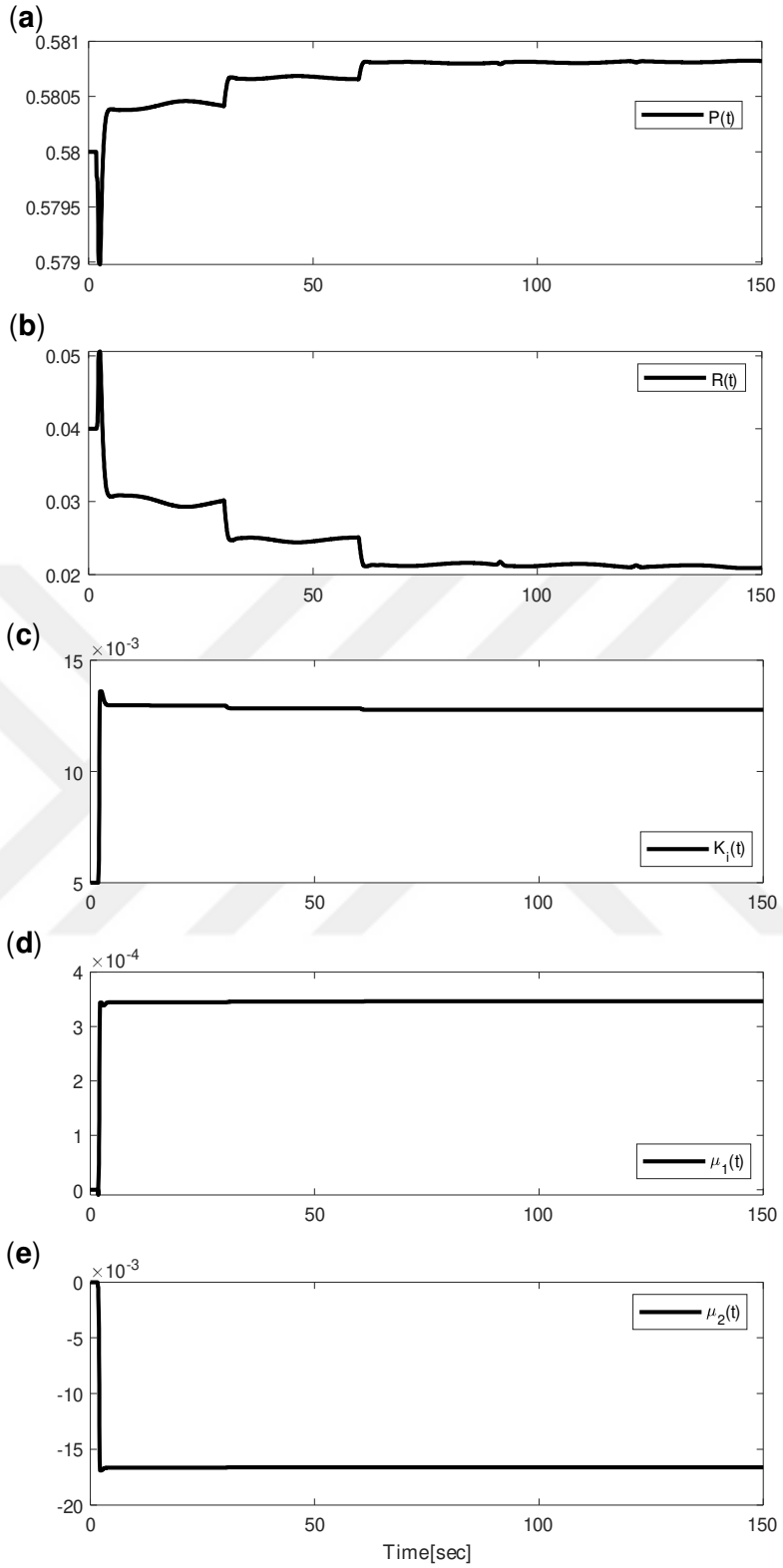

Figure 5.28 : Parametric uncertainty case adaptive parameter results of benchmark system II (CSTR).

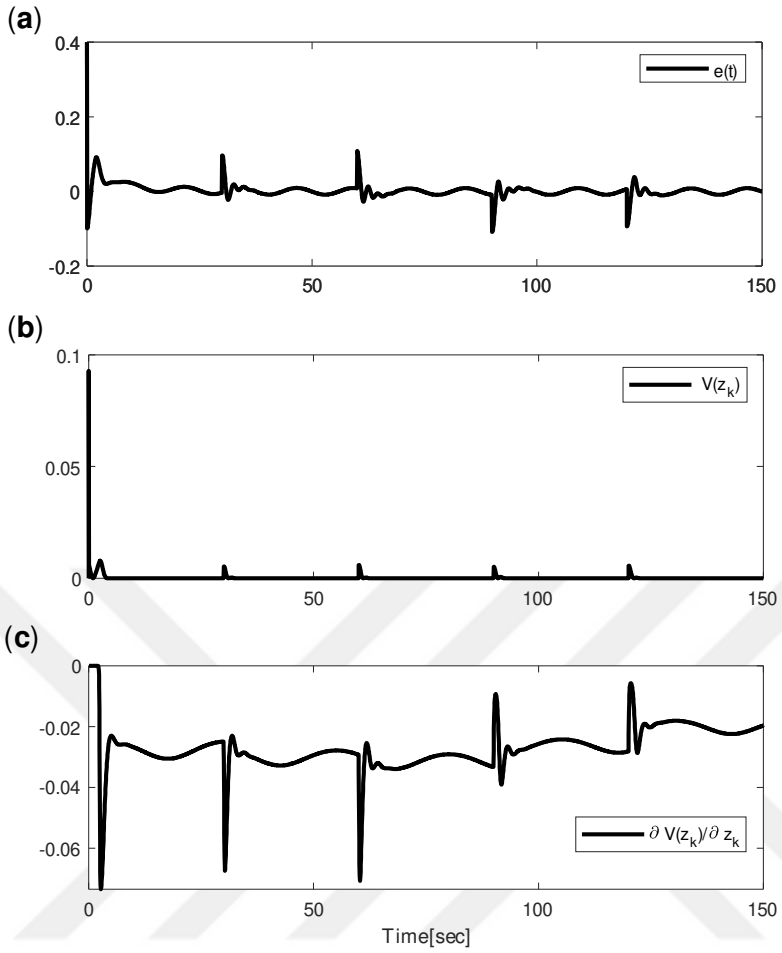

Figure 5.29 : Parametric uncertainty case trajectory tracking error (**a**), stability analysis (**b**), (**c**) of benchmark system II (CSTR).

5.5.3 Comparison Of The Results With PID Controller

A comprehensive comparison has been conducted between the performance of the Adaptive LSSVR-Based Inverse Optimal Controller With the Integrator for Nonlinear Non-Affine Systems proposed in this thesis and a conventional PID controller. This comparison has been carried out in detail for both benchmark systems, including the nominal case and scenarios involving measurement noise, input disturbance, and parametric uncertainty. For a fair comparison, the integral gain calculated by the proposed method has been utilized, while the proportional and derivative parameters of the PID controller have been determined using the Ziegler-Nichols method [128]. The summarized results are provided below in sections 5.5.3.1 and 5.5.3.2.

To enable numerical comparison, the integral square error (ISE) performance index is computed for each simulated case. The ISE integrates the square of the error over time, effectively capturing both large and persistent errors as well as minor errors that persist over extended durations [129]. The ISE criterion is commonly employed to formulate a performance metric for optimal control problems such as the linear quadratic regulator or linear tracking control, while taking into account constraints on the inputs [130, 131]. In this thesis, the ISE has been utilized as a comparison metric.

$$\text{ISE} = \int \mathrm{e}^2(\mathrm{t})\mathrm{dt} \tag{5.67}$$

5.5.3.1 Benchmark system I

Table 5.1 summarizes the performance results for benchmark system I described by (5.61,5.62,5.63). The table includes the ISE performance index for various simulated scenarios, covering the nominal case as well as scenarios involving measurement noise, input disturbance, and parametric uncertainty. It is evident from the table that the proposed control method consistently outperforms the PID controller across all cases. In each scenario, the PID controller's gain parameters remain constant at $K_p = 0.01$, $K_i = 0.0095$, and $K_d = 0.0024$. The integral gain is adjusted to match the value utilized in the proposed control method's integral part, with proportional and derivative gains adapted accordingly. Significantly, the results obtained by the proposed method notably exceed those of the PID controller, particularly in scenarios involving noise, input disturbance, and parametric uncertainty.

Table 5.1 : 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

Nominal case

In Fig. 5.30, the trajectory tracking outcome achieved by the PID controller for benchmark system I under nominal conditions has been depicted. The trajectory tracking performance is illustrated in Fig. 5.30(a), while Fig. 5.30(b) showcases the control signal generated by the PID controller.

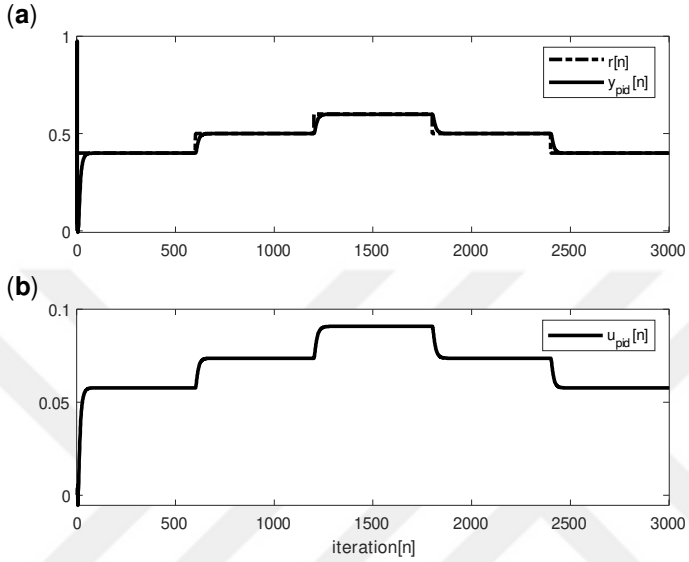

Figure 5.30 : Tracking performance (**a**), PID control signal (**b**) for nominal case

Measurement noise case

In Fig. (5.31)(a), the trajectory tracking performance achieved by the PID controller when measurement noise is introduced is illustrated. Fig. (5.31)(b) presents the control input signal generated by the PID controller under these conditions. A comparison between Fig. (5.31) and Fig. (5.9) clearly demonstrates that the proposed method outperforms the PID controller in handling measurement noise. Despite efforts, the PID control signal struggles to mitigate the impact of the introduced noise.

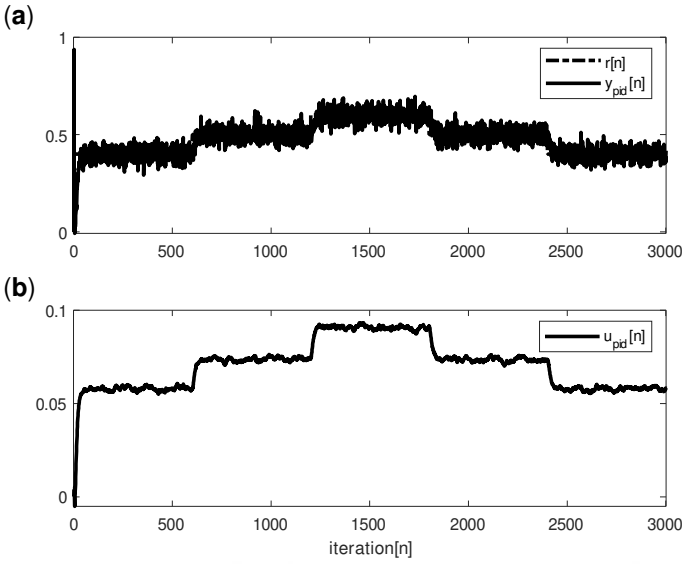

Figure 5.31 : Tracking performance (**a**), PID control signal (**b**) for measurement noise case.

Input disturbance case

Figure (5.32) illustrates the system output alongside the control signal for the case involving an input disturbance of $u_{d_n} = 0.01\sin(\frac{2\pi}{12.5}n)$, when the PID controller is utilized. Upon comparing Figure (5.32) with Figure (5.12), it becomes evident that the proposed method achieves significantly better performance than that obtained by the PID controller.

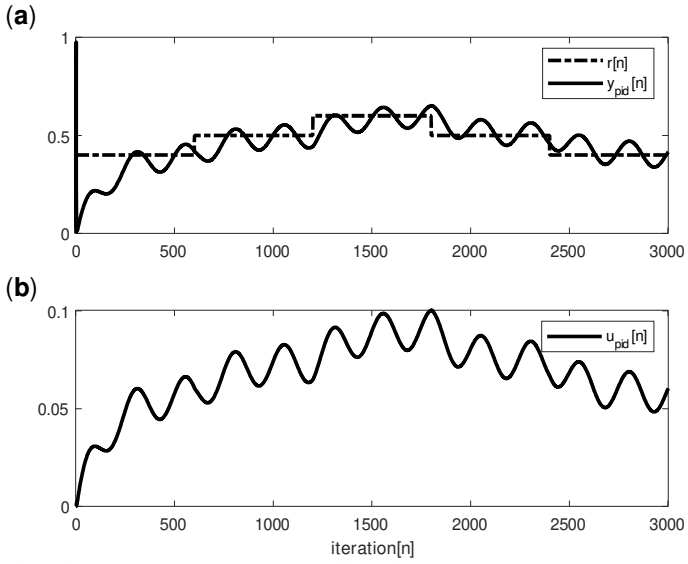

Figure 5.32 : Tracking performance (**a**), PID control signal (**b**) for input disturbance case.

Parametric uncertainty case

Figure (5.33)(a) illustrates the trajectory tracking performance, while Figure (5.33)(b) presents the control input signal generated by the PID controller under the influence of parametric uncertainty, defined as $p_n = 0.1(1 + 0.1\sin(2\pi\frac{1}{25}n))$. Upon comparing Figures (5.15) and (5.33), it is evident that the proposed control method effectively delivers superior tracking performance compared to the PID controller.

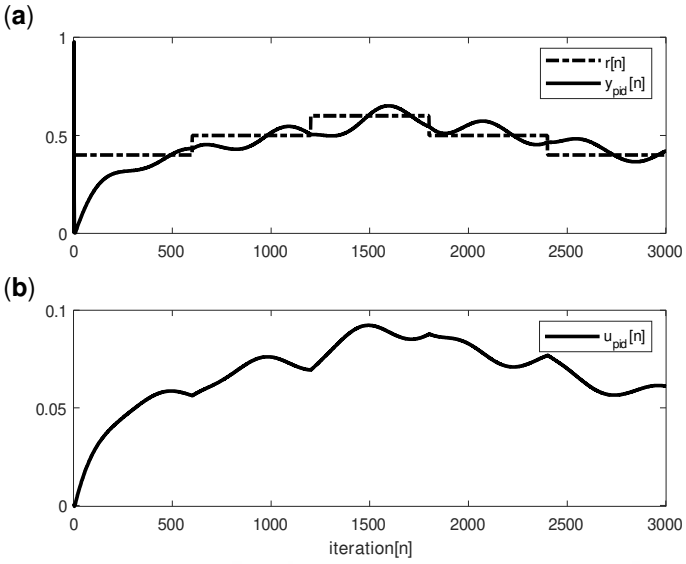

Figure 5.33 : Tracking performance (**a**), PID control signal (**b**)) for parametric uncertainty case

5.5.3.2 Benchmark system II

Table 5.2 presents the computed ISE performance index values for the system described by (5.64,5.65,5.66) across all simulated scenarios. It clearly illustrates the superiority of the proposed control method over the conventional PID controller, both in nominal conditions and when the system is subjected to measurement noise, input disturbance, and parametric uncertainty. The PID controller's gain parameters remain constant across all cases: $K_p = 0.004$, $K_i = 0.0041$, $K_d = 0.001$. For benchmark system II, the integral gain is set equal to that of the proposed controller, with the proportional and derivative gains adjusted accordingly.

Table 5.2 : 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

Nominal case

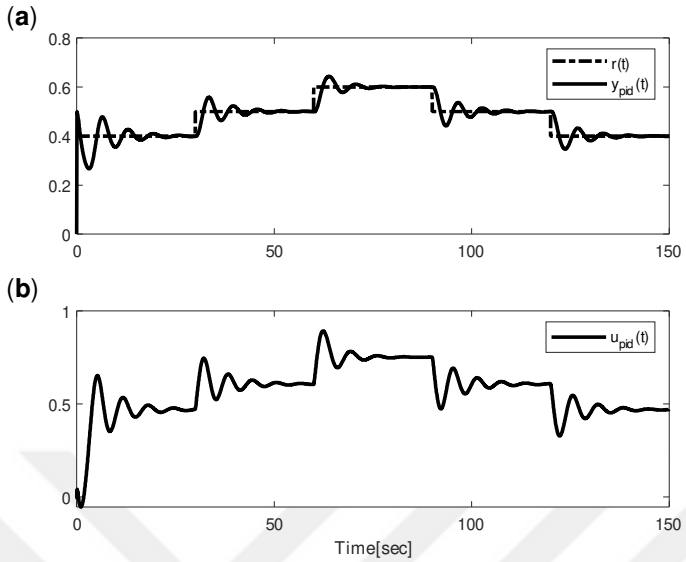

Figure 5.34 : Tracking performance (**a**), PID control signal (**b**)

Figure (5.34) depicts the trajectory tracking performance and the control input signal generated by the PID controller for benchmark system II. A comparison between figures (5.18) and (5.34) reveals that the proposed control methodology achieves successful tracking while effectively eliminating transient effects compared to PID control.

Measurement noise case

Figure (5.35) displays the trajectory tracking results achieved by the PID controller in the presence of a 30 dB measurement noise applied to benchmark system II. Upon comparing figures (5.21) and (5.35), it becomes evident that the proposed control method outperforms PID control. The control input generated by the PID controller struggles to overcome both measurement noise and transient effects.

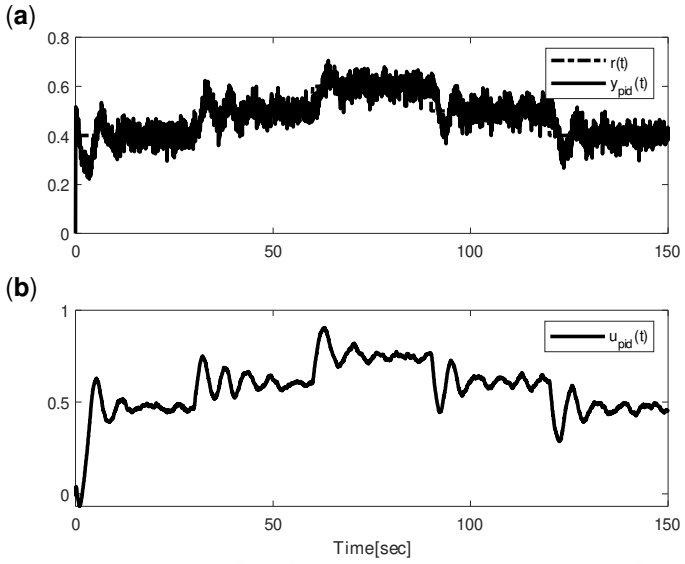

Figure 5.35 : Tracking performance(**a**), PID control signal (**b**) for measurement noise case (CSTR).

Input disturbance case

Figure (5.36) presents the tracking output and control signal obtained for PID control when an input disturbance of $d(t) = 0.01\sin(\frac{2\pi}{12.5}t)$ is applied to benchmark system II. By comparing the results depicted in figures (5.24) and (5.36), it is evident that the proposed control method outperforms PID control. It successfully achieves precise trajectory control while effectively mitigating the effects of the disturbance. In conclusion, the adaptive nature and optimization capability of the proposed control method ensure robust control performance.

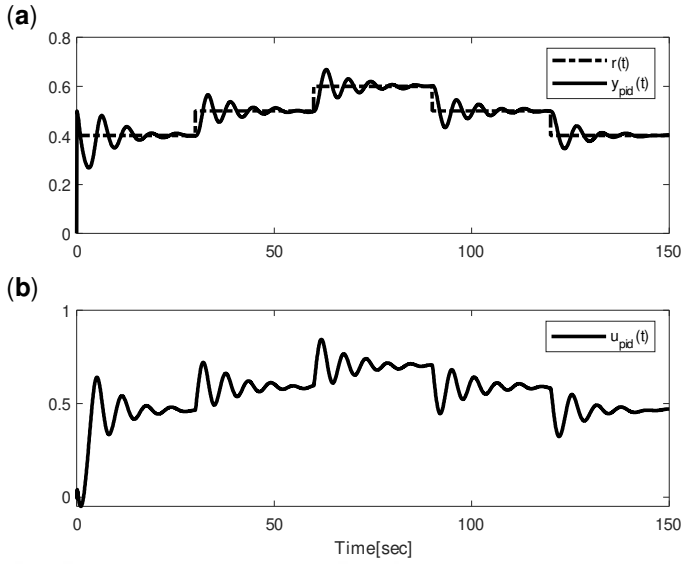

Figure 5.36 : Tracking performance (**a**), PID control signal (**b**) for input disturbance case (CSTR).

Parametric uncertainty case

Figure (5.37) showcases the output trajectory and control input achieved by PID control in the presence of parametric uncertainty within the system. Upon comparing figures (5.27) and (5.37), it is confirmed that the results obtained by the proposed method outperform those obtained by the PID controller.

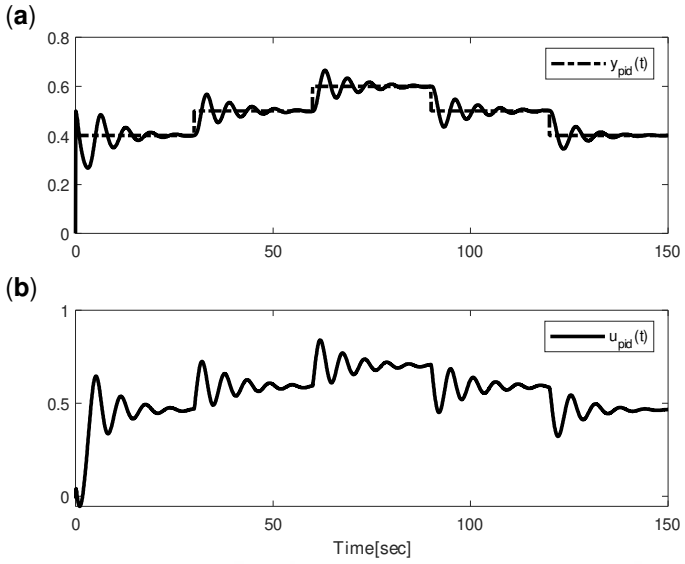

Figure 5.37 : Tracking performance (**a**), PID control signal (**b**) for parametric uncertainty case (CSTR).

The simulation results confirm the efficacy of the proposed control approach in achieving precise trajectory control comparatively. It exhibits rapid attenuation of minor transient disturbances while maintaining low steady-state errors in comparison PID control. The robustness of the method is verified, as it consistently delivers reliable performance even in the presence of factors such as measurement noise, input disturbances, or uncertainties in system parameters.

6. CONCLUSION

When the inverse optimal control technique was first proposed, it offered a novel solution to nonlinear optimal control problems as an alternative to the conventional strategy that relies on the Hamilton-Jacobi-Bellman equation (HJB). The inverse optimal control research in literature has focused on using this technique with affine systems. However, the application of the inverse optimal control technique in non-affine systems has not yet been investigated.

This thesis uses the NN, LSSVR techniques and NARMA-L2 modeling method to present a novel approach to inverse optimal control that is expanded for nonlinear and non-affine systems. Two control architectures have been proposed in this thesis to apply inverse optimal control method for nonlinear and non-affine systems. The first approach introduces a control strategy using neural networks. The key contributions of this approach include obtaining the NARMA-L2 model of a non-affine nonlinear system based on its input-output data using multilayer feedforward neural networks and transforming the non-affine system into an affine model in an offline modeling step. Subsequently, an inverse optimal control law is derived by calculating an adaptive controller parameter using a neural network-based approach, utilizing the affine NARMA-L2 model of the target system. The proposed first approach adopts an adaptive control architecture wherein the controller parameter P is computed online through a recurrent neural network at each time step to minimize a cost function derived from the control error.

The second proposed innovative approach introduces another inverse optimal control method designed for nonlinear and non-affine systems, combining NARMA-L2 modeling and online LSSVR. Initially, the online LSSVR technique is employed to derive the NARX model for non-affine systems, which is subsequently decomposed into NARMA-L2 submodels. This sequential process facilitates the conversion

of a non-affine system model into an affine one. An additional distinctive feature of the proposed method is the online computation of all parameters of the proposed inverse optimal controller with an integrator, this is achieved through the Levenberg-Marquardt algorithm. In summary, the key innovations and contributions of the second work encompass applying the inverse optimal control method to nonlinear non-affine systems through the transformation from a non-affine to an affine system model using the NARMA-L2 modeling approach, incorporating online LSSVR in the design of the inverse optimal controller, and dynamically optimizing the controller parameters of the inverse optimal controller in real-time using the Levenberg-Marquardt algorithm. In order to express the superiority of the proposed controller in comparison to the traditional one of the most common controllers namely PID from the literature, detailed simulation and performance results are also supplied.

In both of the proposed methods, extensive simulation experiments were conducted on various benchmark systems using MATLAB, confirming the superior efficacy and performance of the proposed methodologies under diverse conditions, including scenarios involving measurement noise, parametric uncertainty, and input disturbances.

We hope to further extend the application of the inverse optimal control method to nonlinear non-affine systems in future studies. The integration of multiple machine-learning-based algorithms will be part of this improvement. In addition, our goal is to create new adaptive inverse optimal controller structures that do not require transformation parameters, thereby making the transition from the NARX model to the NARMA-L2 model simpler.

REFERENCES

[1] **Uçak, K. and Günel, G.** (2016). A Novel Adaptive NARMA-L2 Controller Based on Online Support Vector Regression for Nonlinear Systems., *Neural Processing Letters*, *44*(3), 857–886.

[2] **Narendra, K. and Mukhopadhyay, S.** (1997). Adaptive control using neural networks and approximate models, *IEEE Transactions on Neural Networks*, *8*(3), 475–485.

[3] **McShane, E.J.** (1939). On Multipliers for Lagrange Problems, *American Journal of Mathematics*, *61*(4), 809–819, `http://www.jstor.org/stable/2371626`.

[4] **Bliss, G.A.** (1946). *Lectures on the Calculus of Variations*, University of Chicago press.

[5] **Bellman, R.** (1957). A Markovian decision process, *Journal of mathematics and mechanics*, 679–684.

[6] **Pontryagin, L.S.** (2018). *Mathematical theory of optimal processes*, Routledge.

[7] **Letov, A.** (1960). Analytical design of controllers I, *Avtomatika i telemekhanika*, *21*(4), 436–441.

[8] **Rekasius, Z. and Hsia, T.** (1964). On an inverse problem in optimal control, *IEEE Transactions on Automatic Control*, *9*(4), 370–375.

[9] **Akhiezer, N.I.** (1988). *The calculus of variations*, CRC Press.

[10] **Kalman, R.E.** (1964). When Is a Linear Control System Optimal?, *Journal of Basic Engineering*, *86*(1), 51–60.

[11] **Ornelas, F.**, **Sanchez, E.N. and Loukianov, A.G.** (2010). Discrete-time inverse optimal control for nonlinear systems trajectory tracking, *49th IEEE Conference on Decision and Control (CDC)*, pp.4813–4818.

[12] **Shu, H. and Pi, Y.** (2000). PID neural networks for time-delay systems, *Computers & Chemical Engineering*, *24*, 859–862.

[13] **Sanci, M.E.**, **Halis, S. and Kaplan, Y.** (2017). Optimization of machining parameters to minimize surface roughness in the turning of carbon-filled and glass fiber-filled polytetrafluoroethylene, *Materials Design and Applications*, 295–305.

[14] **Onder Efe, M. and Kaynak, O.** (2000). A comparative study of soft-computing methodologies in identification of robotic manipulators, *Robotics and Autonomous Systems*, *30*(3), 221–230.

[15] **Efe, M. and Kaynak, O.** (1999). A comparative study of neural network structures in identification of nonlinear systems, *Mechatronics*, *9*(3), 287–300.

[16] **Gretton, A.**, **Doucet, A.**, **Herbrich, R.**, **Rayner, P. and Scholkopf, B.** (2001). Support vector regression for black-box system identification, *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing*, pp.341–344.

[17] **Rong, H.**, **Zhang, G. and Zhang, C.** (2005). Application of support vector machines to nonlinear system identification, *Proceedings Autonomous Decentralized Systems, 2005. ISADS 2005.*, pp.501–507.

[18] **Suykens, J.** (2001). Nonlinear modelling and support vector machines, *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics*, volume 1, pp.287–294 vol.1.

[19] **Denai, M.**, **Palis, F. and Zeghbib, A.** (2004). ANFIS based modelling and control of non-linear systems : a tutorial, *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, volume 4, pp.3433–3438 vol.4.

[20] **Majstorovic, M.**, **Nikolic, I.**, **Radovic, J. and Kvascev, G.** (2008). Neural network control approach for a two-tank system, *2008 9th Symposium on Neural Network Applications in Electrical Engineering*, pp.215–218.

[21] **Nyandoro, O.T.**, **Pedro, J.O.**, **Dahunsi, O.A. and Dwolatzky, B.** (2011). Linear slip control formulation for vehicular anti-lock braking system with suspension effects, *IFAC Proceedings Volumes*, *44*(1), 4779–4784.

[22] **Jesus, O.D.**, **Pukrittayakamee, A. and Hagan, M.T.** (2001). A comparison of neural network control algorithms, *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*, *1*, 521–526 vol.1.

[23] **Pukrittayakamee, A.**, **Jesus, O.D. and Hagan, M.T.** (2002). Smoothing the control action for NARMA-L2 controllers, *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002.*, *3*, III–III.

[24] **Hagan, M.T.**, **Demuth, H.B. and Jesús, O.D.** (2002). An introduction to the use of neural networks in control systems, *International Journal of Robust and Nonlinear Control*, *12*(11), 959–985.

[25] **Wahyudi**, **Mokri, S.S. and Shafie, A.A.** (2008). Real time implementation of NARMA L2 feedback linearization and smoothed NARMA L2 controls of a single link manipulator, *2008 International Conference on Computer and Communication Engineering*, pp.691–697.

[26] **Akbarimajd, A. and Kia, S.** (2010). NARMA-L2 controller for 2-DoF underactuated planar manipulator, *2010 11th International Conference on Control Automation Robotics and Vision*, pp.195–200.

[27] **Vesselenyi, T.**, **Dzitac, S.**, **Dzitac, I. and Manolescu, M.J.** (2007). Fuzzy and Neural Controllers for a Pneumatic Actuator, *International Journal of Computers, Communications and Control (IJCCC)*, *2*, 375–387.

[28] **Ioannou, P.A.** (2001). Adaptive control of systems with actuator and sensor nonlinearities: Gang Tao and Petar V. Kokotovic; Wiley, New York, 1996, ISBN: 0-471-15654-X, *Autom.*, *37*, 1685–1687.

[29] **Chen, T.**, **Zhang, S. and Ge, S.S.** (2021). UDE-based Robust Control for a Class of Nonaffine Nonlinear Systems with Uncertainties and Output Constraint, *2021 40th Chinese Control Conference (CCC)*, pp.247–252.

[30] **Zhang, Q. and Wang, C.** (2017). Robust adaptive backstepping control for a class of constrained non-affine nonlinear systems via self-organizing Hermite-polynomial-based neural network disturbance observer, *Advances in Mechanical Engineering*, *9*(5), 1687814017702811.

[31] **Bu, X.**, **Jiang, B. and Lei, H.** (2023). Performance Guaranteed Finite-Time Non-Affine Control of Waverider Vehicles Without Function-Approximation, *IEEE Transactions on Intelligent Transportation Systems*, *24*(3), 3252–3262.

[32] **Li, G.**, **Liu, Y.**, **Li, Y. and Bu, X.** (2020). Adaptive back-stepping control of high-order uncertain nonlinear systems that a funnel control scheme with uncertain dynamics, *2020 International Conference on Electrical Engineering and Control Technologies (CEECT)*, pp.1–8.

[33] **Li, Y.**, **Zhu, Q.**, **Zhang, J. and Deng, Z.** (2022). Adaptive Fixed-Time Neural Networks Control for Pure-Feedback Non-Affine Nonlinear Systems with State Constraints, *Entropy*, *24*, 737.

[34] **Son, T.D. and Nguyen, Q.** (2019). Safety-Critical Control for Non-affine Nonlinear Systems with Application on Autonomous Vehicle, *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp.7623–7628.

[35] **Zhao, S.**, **Pan, Y.**, **Du, P. and Liang, H.** (2020). Adaptive control for non-affine nonlinear systems with input saturation and output dead zone, *Applied Mathematics and Computation*, *386*, 125506, `https://www.sciencedirect.com/science/article/pii/S0096300320304641`.

[36] **Al-Tamimi, A.** (2012). Optimal controller design algorithm for non-affine in input discrete-time nonlinear system, *Jordan Journal of Mechanical and Industrial Engineering*, *6*, 155–161.

[37] **Zhao, T. and Sui, S.** (2006). Adaptive Control for a Class of Non-affine Nonlinear Systems via Two-Layer Neural Networks, *2006 6th World Congress on Intelligent Control and Automation*, volume 1, pp.958–962.

[38] **Wang, H.**, **Tian, Y. and Vasseur, C.** (2015). Non-Affine Nonlinear Systems Adaptive Optimal Trajectory Tracking Controller Design and Application, *Studies in Informatics and Control*, *21*, 5–11.

[39] **Lin, N; Chi, R. and Huang, B.** (2023). Data-driven set-point control for nonlinear nonaffine systems, *Information Sciences*, *625*, 237–254.

[40] **Kim, T.**, **Park, G.**, **Kwak, K.**, **Bae, J. and Lee, W.** (2022). Smooth Model Predictive Path Integral Control Without Smoothing, *IEEE Robotics and Automation Letters*, *7*(4), 10406–10413.

[41] **Nekoo, S.R. and Geranmehr, B.** (2014). Nonlinear observer-based optimal control using the state-dependent Riccati equation for a class of non-affine control systems, *Journal of Control Engineering and Applied Informatics*, *16*, 5–13.

[42] **Zhang, X. and Hou, Z.** (2023). Data-driven predictive point-to-point iterative learning control, *Neurocomputing*, *518*, 431–439.

[43] **Thierry-Marie Guerra, Anh-Tu Nguyen, M.D.** (2019). Control of SISO non-affine-in-control discrete-time systems using Takagi-Sugeno models, *IFAC-PapersOnLine*, *52*(11), 79–84, 5th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2019.

[44] **Bu, X.**, **Jiang, B. and Lei, H.** (2022). Nonfragile Quantitative Prescribed Performance Control of Waverider Vehicles With Actuator Saturation, *IEEE Transactions on Aerospace and Electronic Systems*, *58*(4), 3538–3548.

[45] **Bu, X. and Qi, Q.** (2022). Fuzzy Optimal Tracking Control of Hypersonic Flight Vehicles via Single-Network Adaptive Critic Design, *IEEE Transactions on Fuzzy Systems*, *30*(1), 270–278.

[46] **Bu, X.** (2019). Actor-Critic Reinforcement Learning Control of Non-Strict Feedback Nonaffine Dynamic Systems, *IEEE Access*, *7*, 65569–65578.

[47] **Bai, W.**, **Liu, P.X. and Wang, H.** (2022). Neural-Network-Based Adaptive Fixed-Time Control for Nonlinear Multiagent Non-Affine Systems, *IEEE Transactions on Neural Networks and Learning Systems*, 1–14.

[48] **Ge, S.**, **Zhang, J. and Lee, T.** (2004). Adaptive MNN control for a class of non-affine NARMAX systems with disturbances, *Systems and Control Letters*, *53*(1), 1–12, https://www.sciencedirect.com/science/article/pii/S0167691104000337.

[49] **P. Bagheri, L. Behjat, Q.S.** (2022). Nonlinear control of a class of non-affine variable-speed variable-pitch wind turbines with radial-basis function neural networks, *ISA Transactions*, *131*, 197–209.

[50] **Bu, X.** (2018). Air-Breathing Hypersonic Vehicles Funnel Control Using Neural Approximation of Non-affine Dynamics, *IEEE/ASME Transactions on Mechatronics*, *23*(5), 2099–2108.

[51] **Bu, X.**, **Hua, C.**, **Lv, M. and Wu, Z.** (2023). Flight Control of Waverider Vehicles with Fragility-avoidance Prescribed Performance, *IEEE Transactions on Aerospace and Electronic Systems*, 1–15.

[52] **Büchi, R.** (2010). *State space control, LQR and observer : step by step introduction with Matlab examples*, Norderstedt Books on Demand, Norderstedt, 1. edition, https://digitalcollection.zhaw.ch/handle/11475/3792.

[53] **Mainprice, J.**, **Hayne, R. and Berenson, D.** (2016). Goal Set Inverse Optimal Control and Iterative Replanning for Predicting Human Reaching Motions in Shared Workspaces, *IEEE Transactions on Robotics*, *32*(4), 897–908.

[54] **Lopez-Franco, C.**, **López-Franco, M.**, **Alanis, A.**, **Gómez-Avila, J. and Arana-Daniel, N.** (2015). Real-Time Inverse Optimal Neural Control for Image Based Visual Servoing with Nonholonomic Mobile Robots, *Mathematical Problems in Engineering*, *2015*, 1–12.

[55] **Gurubel, K.**, **Sanchez, E.**, **Coronado, A.**, **Zúñiga Grajeda, V.**, **Sulbaran, B. and Breton-Deval, L.** (2019). Inverse optimal neural control via passivity approach for nonlinear anaerobic bioprocesses with biofuels production, *Optimal Control Applications and Methods*, *40*.

[56] **Chan, V.M.**, **Hernández-Vargas, E.A. and Sánchez, E.N.** (2021). Neural inverse optimal control applied to design therapeutic options for patients with COVID-19, *2021 International Joint Conference on Neural Networks (IJCNN)*, pp.1–7.

[57] **Lopez-Garcia, T.B.**, **Sanchez, E.N. and Ruiz-Cruz, R.** (2019). Real-time implementation of battery bank charge–discharge based on neural inverse optimal control, *IET Renewable Power Generation*, *13*(16), 3124–3132.

[58] **Leon, B.S.**, **Alanis, A.Y.**, **Sanchez, E.N.**, **Ornelas-Tellez, F. and Ruiz-Velazquez, E.** (2014). Neural Inverse Optimal Control via Passivity for Subcutaneous Blood Glucose Regulation in Type 1 Diabetes Mellitus Patients, *Intelligent Automation and Soft Computing*, *20*, 279–295.

[59] **Ruiz-Cruz, R.**, **Sanchez, E.N.**, **Ornelas-Tellez, F.**, **Loukianov, A.G. and Harley, R.G.** (2013). Particle Swarm Optimization for Discrete-Time Inverse Optimal Control of a Doubly Fed Induction Generator, *IEEE Transactions on Cybernetics*, *43*(6), 1698–1709.

[60] **Ulusoy, L.**, **Güzelkaya, M. and Eksin.İ** (2019). Fusion of inverse optimal and model predictive control strategies, *Transactions of the Institute of Measurement and Control*, *42*, 1122–1134.

[61] **Fotouhi, R. and Pourgholi, M.** (2021). Discrete-time Inverse Optimal Control for Consensus of Multi-Agent Systems via a Novel Meta-Heuristic Algorithm, *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, pp.1–5.

[62] **Perez-Villalpando, M.**, **Tun, K.**, **Muro, C. and Fausto, F.** (2021). Inverse Optimal Control Using Metaheuristics of Hydropower Plant Model via Forecasting Based on the Feature Engineering, *Energies*, *14*, 7356.

[63] **Ricalde, L.J. and Sanchez, E.** (2012). Inverse optimal neural control of a class of nonlinear systems with constrained inputs for trajectory tracking, *Optimal Control Applications and Methods*, *33*.

[64] **Vega, C.J.**, **Suarez, O.J.**, **Sanchez, E.N.**, **Chen, G.**, **Elvira-Ceja, S. and Rodriguez, D.I.** (2020). Trajectory Tracking on Uncertain Complex Networks via NN-Based Inverse Optimal Pinning Control, *IEEE Transactions on Neural Networks and Learning Systems*, *31*(3), 854–864.

[65] **Rios, J.**, **Alanis, A.**, **Lopez-Franco, M.**, **Lopez-Franco, C. and Arana-Daniel, N.** (2017). Real-time neural identification and inverse optimal control for a tracked robot, *Advances in Mechanical Engineering*, *9*, 168781401769297.

[66] **Lopez, V.**, **Sanchez, E.**, **Alanis, A. and Rios, J.** (2016). Real-Time Neural Inverse Optimal Control for a Linear Induction Motor, *International Journal of Control*, *90*, 1–29.

[67] **Hernandez-Mejia, G.**, **Alanis, A. and Hernandez Vargas, E.A.** (2018). Neural inverse optimal control for discrete-time impulsive systems, *Neurocomputing*, *314*.

[68] **Villaseñor, C.**, **Rios, J.D.**, **Arana-Daniel, N.**, **Alanis, A.Y.**, **Lopez-Franco, C. and Hernandez-Vargas, E.A.** (2018). Germinal Center Optimization Applied to Neural Inverse Optimal Control for an All-Terrain Tracked Robot, *Applied Sciences*, *8*(1).

[69] **Fotouhi, R. and Pourgholi, M.** (2021). Discrete-time Inverse Optimal Control for Consensus of Multi-Agent Systems via a Novel Meta-Heuristic Algorithm, *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, pp.1–5.

[70] **Fotouhi, R. and Pourgholi, M.** (2021). Discrete-time Inverse Optimal Control for Consensus of Multi-Agent Systems via a Novel Meta-Heuristic Algorithm, *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*, pp.1–5.

[71] **Cai, X.**, **Lin, C.**, **Liu, L. and Zhan, X.** (2018). Inverse optimal control for strict-feedforward nonlinear systems with input delays, *International Journal of Robust and Nonlinear Control*, *28*(8), 2976–2995.

[72] **Lin, Z.**, **Liu, Z.**, **Zhang, Y. and Chen, C.P.** (2022). Adaptive neural inverse optimal tracking control for uncertain multi-agent systems, *Information Sciences*, *584*, 31–49.

[73] **Li, Y.m.**, **Min, X. and Tong, S.** (2020). Adaptive Fuzzy Inverse Optimal Control for Uncertain Strict-Feedback Nonlinear Systems, *IEEE Transactions on Fuzzy Systems*, *28*(10), 2363–2374.

[74] **Li, Y.**, **Min, X. and Tong, S.** (2021). Observer-Based Fuzzy Adaptive Inverse Optimal Output Feedback Control for Uncertain Nonlinear Systems, *IEEE Transactions on Fuzzy Systems*, *29*(6), 1484–1495.

[75] **Min, X.**, **Li, Y. and Tong, S.** (2020). Adaptive Fuzzy Output Feedback Inverse Optimal Control for Vehicle Active Suspension Systems, *Neurocomputing*, *403*.

[76] **Lin, Z.**, **Liu, Z.**, **Zhang, Y. and Chen, C.P.** (2022). Adaptive neural inverse optimal tracking control for uncertain multi-agent systems, *Information Sciences*, *584*, 31–49.

[77] **Carrasco-Gutierrez, C.E. and Sosa, W.** (2019). A DISCRETE DYNAMICAL SYSTEM AND ITS APPLICATIONS, *Pesquisa Operacional*, *39*, 457–469.

[78] **Sanchez, E.N.** (2018). *Discrete-Time Recurrent Neural Control*, CRC Press.

[79] **Galor, O.** (2007). *Discrete Dynamical Systems*, Springer Berlin Heidelberg.

[80] **Lewis, F. and Syrmos, V.** (1995). *Optimal Control*, A Wiley-interscience publication, Wiley.

[81] **Kirk, D.E.** (2004). *Optimal control theory: an introduction*, Courier Corporation.

[82] **Moulay, E. and Perruquetti, W.** (2005). Stabilization of nonaffine systems: a constructive method for polynomial systems, *IEEE Transactions on Automatic Control*, *50*, 520–526.

[83] **Wang, H.**, **Tian, Y. and Vasseur, C.** (2015). Non-Affine Nonlinear Systems Adaptive Optimal Trajectory Tracking Controller Design and Application, *Studies in Informatics and Control*, *21*, 5–11.

[84] **Chien, Y.H.**, **Wang, W.Y.**, **Leu, Y.G. and Lee, T.T.** (2011). Robust Adaptive Controller Design for a Class of Uncertain Nonlinear Systems Using Online T–S Fuzzy-Neural Modeling Approach, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*(2), 542–552.

[85] **Gao, Q. and Gao, Q.** (2017). Universal fuzzy models and universal fuzzy controllers for non-affine nonlinear systems, *Universal fuzzy controllers for non-affine nonlinear systems*, 19–43.

[86] **Labiod, S. and Guerra, T.M.** (2010). Indirect adaptive fuzzy control for a class of nonaffine nonlinear systems with unknown control directions, *International journal of control, Automation and systems*, *8*, 903–907.

[87] **Tombul, G.S.**, **Banks, S.P. and Akturk, N.** (2009). Sliding mode control for a class of non-affine nonlinear systems, *Nonlinear Analysis: Theory, Methods & Applications*, *71*(12), e1589–e1597.

[88] **Krstic, M.**, **Kanellakopoulos, I. and Kokotovic, P.** (1994). Nonlinear design of adaptive controllers for linear systems, *IEEE Transactions on Automatic Control*, *39*(4), 738–752.

[89] **Marino, R. and Tomei, P.** (1996). *Nonlinear Control Design: Geometric, Adaptive and Robust*, Prentice Hall International (UK) Ltd., GBR.

[90] **Sastry, S. and Isidori, A.** (1989). Adaptive control of linearizable systems, *IEEE Transactions on Automatic Control*, *34*, 1123–1131.

[91] **Artstein, Z.** (1983). Stabilization with relaxed controls, *Nonlinear Analysis: Theory, Methods and Applications*, *7*(11), 1163–1173, `https://www.sciencedirect.com/science/article/pii/0362546X83900494`.

[92] **Sontag, E.D.** (1983). A Lyapunov-Like Characterization of Asymptotic Controllability, *SIAM Journal on Control and Optimization*, *21*, 462–471.

[93] **Moulay, E. and Perruquetti, W.** (2005). Stabilization of nonaffine systems: A constructive method for polynomial systems, *IEEE Transactions on Automatic Control*, *50*(4), 520–526.

[94] **Shiriaev, A.S. and Fradkov, A.L.** (2000). Stabilization of invariant sets for nonlinear non-affine systems, *Automatica*, *36*(11), 1709–1715.

[95] **Sanci, M.E.**, **Uçak, K. and Günel.G.Ö** (2023). A Novel Adaptive LSSVR-Based Inverse Optimal Controller With Integrator for Nonlinear Non-Affine Systems, *IEEE Access*, *11*, 137548–137576.

[96] **Sancı, M.E. and Günel, G.** (2024). Neural Network Based Adaptive Inverse Optimal Control for Non-Affine Nonlinear Systems, *Neural Processing Letters*, *56*.

[97] **Middletone, R. and Goodwin, G.C.** (1986). Adaptive computed torque control for rigid link manipulators, *1986 25th IEEE Conference on Decision and Control*, 68–73.

[98] **da Silva, P.S.P. and Batista, S.** (2010). On state representations of nonlinear implicit systems, *International Journal of Control*, *83*(3), 441–456.

[99] **Song, Y.D. and Song, Q.** (2011). Survey of the latest developments in control of non-affine systems, *Proceedings of the 30th Chinese Control Conference*, pp.785–790.

[100] **Kirk, D.** (2004). *Optimal Control Theory: An Introduction*, Dover Books on Electrical Engineering Series, Dover Publications, `https://books.google.com/books?id=fCh2SAtWIdwC`.

[101] **Sepulchre, R.**, **Jankovic, M. and Kokotovic, P.V.** (2011). *Constructive Nonlinear Control*, Springer London.

[102] **Kalman, R.E.** (1964). When Is a Linear Control System Optimal?, *Journal of Basic Engineering*, *86*(1), 51–60, `https://doi.org/10.1115/1.3653115`.

[103] **Freeman, R.A. and Kokotovic, P.V.** (1996). Inverse Optimality in Robust Stabilization, *SIAM Journal on Control and Optimization*, *34*(4), 1365–1391.

[104] **Leon, B.S.**, **Alanis, A.Y.**, **Sanchez, E.N.**, **Ornelas-Tellez, F. and Ruiz-Velazquez, E.**, (2012), Neural inverse optimal control applied to type 1 diabetes mellitus patients, `https://ieeexplore.ieee.org/document/6180310`.

[105] **Sanchez, E.N. and Ornelas-Tellez, F.** (2017). *Discrete-Time Inverse Optimal Control for Nonlinear Systems*, CRC Press.

[106] **Leon, B.**, **Naumova, V.**, **Ruiz-Velazquez, E.**, **McCulloch, A.D. and Sanchez, E.** (2016). Combination of Neural Inverse Optimal Control with a Kernel-Based Regularization Learning Algorithm to Prevent Hypoglycemia in Type 1 Diabetes Patients, *IEEE Transactions on Neural Networks and Learning Systems*.

[107] **Atkinson, C. and Osseiran, A.** (2011). Discrete-space time-fractional processes, *Fractional Calculus and Applied Analysis*, *14*, 201–232.

[108] **Edgar, S.** (2018). *Discrete-Time Recurrent Neural Control : Analysis and Applications.*, CRC Press Taylor and Francis Group, Boca Raton.

[109] **Rios, Y.Y.**, **García-Rodríguez, J.A.**, **Sánchez, O.D.**, **Sanchez, E.N.**, **Alanis, A.Y.**, **Ruiz-Velázquez, E. and Arana-Daniel, N.** (2018). Inverse Optimal Control Using A Neural Multi-Step Predictor For T1DM Treatment, *2018 International Joint Conference on Neural Networks (IJCNN)*, pp.1–8.

[110] **Almobaied, M.**, **Eksin, I. and Guzelkaya, M.** (2019). Inverse Optimal Controller Design Based on Multi-Objective Optimization Criteria for Discrete-Time Nonlinear Systems, *2019 IEEE 7th Palestinian International Conference on Electrical and Computer Engineering (PICECE)*, pp.1–6.

[111] **Almobaied, M.**, **Eksin, I. and Guzelkaya, M.** (2018). Inverse optimal controller based on extended Kalman filter for discrete-time nonlinear systems, *Optimal Control Applications and Methods*, *39*(1), 19–34.

[112] **Akoum, M. and Günel, G.Ö.** (2021). Inverse Optimal Control Based on Improved Grey Wolf Optimization Algorithm, *2021 13th International Conference on Electrical and Electronics Engineering (ELECO)*, pp.47–51.

[113] **Vapnik, V.N.** (1995). *The nature of statistical learning theory*, Springer-Verlag New York, Inc.

[114] **Smola, A. and Schölkopf, B.** (2004). A tutorial on support vector regression, *Statistics and Computing*, *14*, 199–222.

[115] **Suykens, J.** (2001). Nonlinear modelling and support vector machines, *IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188)*, volume 1, pp.287–294 vol.1.

[116] **Gunn, S.R.** *et al.* (1998). Support vector machines for classification and regression, *ISIS technical report*, *14*(1), 5–16.

[117] **Ucak, K. and Oke, G.** (2011). An improved adaptive PID controller based on online LSSVR with multi RBF kernel tuning, *Lecture Notes in Computer Science*, *6943*.

[118] **Ucak, K. and Öke, G.** (2011). RBF neural network controller based on OLSSVR., *ICAIS*, volume6943 of *Lecture Notes in Computer Science*, Springer, pp.40–51.

[119] **Şen, G.D. and Günel, G.Ö.** (2022). A NARMA-L2 Controller Based on Online LSSVR for Nonlinear Systems, ***E. Zattoni, S. Simani and G. Conte**, editors, 15th European Workshop on Advanced Control and Diagnosis (ACD 2019)*, Springer International Publishing, Cham, pp.213–231.

[120] **Iplikci, S.** (2009). Controlling the experimental three-tank system via support vector machines, *Adaptive and Natural Computing Algorithms: 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers 9*, Springer, pp.391–400.

[121] **Wanfeng, S.**, **Shengdun, Z. and Yajing, S.** (2008). Adaptive PID Controller Based on Online LSSVM Identification, *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp.694–698.

[122] **fei Zhu, Y. and yuan Mao, Z.** (2004). Online optimal modeling of LS-SVM based on time window, *2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT '04.*, volume 3, pp.1325–1330 Vol. 3.

[123] **Uçak, K.** (2020). A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems, *Neural Processing Letters*, *51*, 1789–1833.

[124] **Iplikci, S.** (2009). A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems, *International Journal of Robust and Nonlinear Control*, *20*, 1483 – 1501.

[125] **Show, K.Y. and Lee, D.J.**, (2013). Chapter 13 - Bioreactor and Bioprocess Design for Biohydrogen Production, **A. Pandey**, **J.S. Chang**, **P.C. Hallenbecka and C. Larroche**, editors, Biohydrogen, Elsevier, Amsterdam, pp.317–337.

[126] **Wu, W. and Chou, Y.S.** (2010). Adaptive feedforward and feedback control of non-linear time-varying uncertain systems, *International Journal of Control*, *72*, 1127–1138.

[127] **Miller, W.T.**, **Sutton, R.S. and Werbos, P.J.**, (1995). A Bioreactor Benchmark for Adaptive Network-based Process Control, MIT press, pp.387–402.

[128] **Ogata, K.** (2010). *Modern Control Engineering*, Instrumentation and controls series, Prentice Hall.

[129] **Krishnan, K. and Karpagam, G.** (2014). Comparison of PID controller tuning techniques for a FOPDT system, *International Journal of Current Engineering and Technology*, *4*(4), 2667–2670.

[130] **Domanski, P.** (2020). *Control Performance Assessment: Theoretical Analyses and Industrial Practice*, Springer.

[131] **Walton, K. and Gorecki, H.** (1984). On the Evaluation of Cost Functionals, with Particular Emphasis on Time-delay Systems, *IMA Journal of Mathematical Control and Information*, *1*(3), 283–306, `https://doi.org/10.1093/imamci/1.3.283`.

CURRICULUM VITAE

Name SURNAME:
Muhammet Emre SANCI

EDUCATION:

- **B.Sc.:** 2011, Abant Izzet Baysal, Science Faculty, Physics
- **M.Sc.:** 2015, Pamukkale, Engineering Faculty, Electrics And Electronics Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2011 Honor Student Abant Izzet Baysal Univeristy Physics.
- 2013 Research Assitant at Mechatronics Engineering, Pamukkale University
- 2016 Completed Msc Pamukkale University for Electrics And Electronics Engineering

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Sanci M. E.** , **Uçak K**,**Günel G. O.** Novel Adaptive LSSVR-Based Inverse Optimal Controller With Integrator for Nonlinear Non-Affine Systems, *IEEE Access* , vol. 11, pp. 137548-137576, 2023, doi: 10.1109/ACCESS.2023.3339571.
- **Sanci M. E.** ,**Günel G. O.** Neural Network Based Adaptive Inverse Optimal Control for Non-Affine Nonlinear Systems , *Neural Processing Letters* , vol. 56:46, 2024 ,doi:/10.1007/s11063-024-11456-9