M. UĞUR SOYDEMİR

IZMIR KATIP CELEBI UNIVERSITY

2020

IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CONTROL OF ROTARY INVERTED PENDULUM SYSTEM WITH LEARNING FEEDBACK LINEARIZATION BASED STABLE ROBUST ADAPTIVE CONTROLLER

M.Sc. THESIS

Mehmet Uğur SOYDEMİR

Department of Electrical and Electronics Engineering

Thesis Advisor: Assoc. Prof. Savaş ŞAHİN

JANUARY 2020

IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CONTROL OF ROTARY INVERTED PENDULUM SYSTEM WITH LEARNING FEEDBACK LINEARIZATION BASED STABLE ROBUST ADAPTIVE CONTROLLER

M.Sc. THESIS

Mehmet Uğur SOYDEMİR
(Y180230001)

Department of Electrical and Electronics Engineering

Thesis Advisor: Assoc. Prof. Savaş ŞAHİN

JANUARY 2020

İZMİR KATİP CELEBİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ÖĞRENEN GERİ BESLEMELİ DOĞRUSALLAŞTIRMA TABANLI KARARLI GÜRBÜZ UYARLANIR DENETLEYİCİ İLE DÖNEL TERS SARKAÇ SİSTEMİNİN KONTROLÜ

YÜKSEK LİSANS TEZİ

Mehmet Uğur SOYDEMİR

(Y180230001)

Elektrik Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Savaş ŞAHİN

OCAK 2020

Mehmet Uğur SOYDEMİR, a **M.Sc.** student of **IKCU Graduate School Of Natural And Applied Sciences**, successfully defended the thesis entitled **CONTROL OF ROTARY INVERTED PENDULUM SYSTEM WITH LEARNING FEEDBACK LINEARIZATION BASED STABLE ROBUST ADAPTIVE CONTROLLER**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :

Assoc. Prof. Savaş ŞAHİN

İzmir Kâtip Çelebi University ....

Jury Members :

Assoc. Prof. Levent ÇETİN

İzmir Kâtip Çelebi University ....

Assoc. Prof Selami BEYHAN

İzmir Demokrasi University ....

Date of Defense **: 30.12.2019**

To my family

FOREWORD

Firstly, I would like to give thank my thesis supervisor Assoc. Prof. Savaş ŞAHİN who does not spare his help, support, patience and knowledge to me and he shed light on me since I started my academic career. Secondly, I owe Parvin Bulucu, Alkım Gökçen, Tuğçe Toprak, Levent Taşdöğen and Rıza Gökbunar a great debt of gratitude for their contributions to my study. The Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant 116E170 supported this thesis.

January 2020

Mehmet Uğur SOYDEMİR

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABBREVIATIONS

ARMA	**:** Auto-regressive Moving Average
ANN	**:** Artificial Neural Network
LFL	**:** Learning Feedback Linearization
MLP	**:** Multi-Layer Perceptron
MSE	**:** Mean Square Error
NARMA	**:** Nonlinear Autoregressive Moving Average
PD	**:** Proportional-Derivative
ROTPEN	**:** Rotary Inverted Pendulum
SIMO	**:** Single Input Multi Output
SISO	**:** Single Input Single Output

CONTROL OF ROTARY INVERTED PENDULUM SYSTEM WITH LEARNING FEEDBACK LINEARIZATION BASED STABLE ROBUST ADAPTIVE CONTROLLER

ABSTRACT

This thesis presents a learning feedback linearization (LFL) based stable robust adaptive controller design for a rotary inverted pendulum (ROTPEN) plant. The proposed adaptive controller design algorithm is based on a linear controller model and a feedback linearized plant model obtained from a nonlinear auto-regressive moving-average (NARMA) based LFL. The proposed algorithm is achieved by three progressive stages as follows; i) NARMA based LFL is used to obtain a feedback linearized model for a nonlinear plant by using the artificial neural network (ANN), ii) the NARMA-LFL based plant might be identified as an auto-regressive moving average (ARMA) plant model, and iii) the closed-loop control system providing Schur stability conditions is constituted by both ARMA plant and controller models. Once the training phase of ANN is fulfilled, the feedback linearized nonlinear plant might be identified as the ARMA model including the combination of the nonlinear plant and it's learned LFL block. The proposed stable robust adaptive control algorithm is implemented via the ARMA models of both the plant and the controller provided the Schur stability conditions for the overall closed-loop system. Robustness properties of both the linearized plant model and the overall closed-loop system are employed with the ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ defined as the identification error of the linearized nonlinear plant and the tracking error, respectively. In conclusion, the proposed LFL-based-stable-adaptive-controller is applied for ROTPEN model and its physical experimental setup. The performance of the proposed controller is compared with the Proportional-Derivative controller in terms of mean square error for tracking error.

ÖĞRENEN GERİ BESLEMELİ DOĞRUSALLAŞTIRMA TABANLI KARARLI, GÜRBÜZ, UYARLANIR KONTROLÖR İLE DÖNEL TERS SARKAÇ SİSTEMİNİN KONTROLÜ

ÖZET

Bu tez, öğrenen geri beslemeli doğrusallaştırma (ÖGD) tabanlı kararlı, gürbüz, uyarlanır bir kontrolör tasarımını bir dönel ters sarkaç (DTS) sistemi için sunmaktadır. Önerilen uyarlanır kontrolör algoritması bir doğrusal kontrolör modeline ve bir doğrusal olmayan özyinelemeli kayan-ortalama (DÖKO) tabanlı ÖGD ile elde edilen geri beslemeli doğrusallaştırılmış sistem modeline dayanmaktadır. Önerilen algoritma 3 aşamadan oluşmaktadır; i) DÖKO tabanlı ÖGD, bir yapay sinir ağı (YSA) yardımıyla doğrusal olmayan sistem için bir geri beslemeli doğrusallaştırılmış model elde etmede kullanılması, ii) DÖKO-ÖGD tabanlı sistemin bir özyinelemeli kayan ortalama (ÖKO) sistem modeli ile tanılanabilmesi ve iii) Schur kararlılık koşullarını sağlayan kapalı çevrim kontrol sistemini ÖKO sistem ve kontrolör modelleri tarafından oluşturulması. Bir kere YSA'nın eğitim aşaması yerine getirildiğinde, geri beslemeli doğrusallaştırılmış doğrusal olmayan sistem, doğrusal olmayan sistemin ve onun öğrenilmiş ÖGD bloğunun bir kombinasyonunu içeren bir ÖKO modeli olarak tanılanmaktadır. Önerilen kararlı, gürbüz, uyarlanır kontrol algoritması hem sistemin hem de tüm kapalı çevrim sistem için Schur kararlılık koşullarını sağlayan ÖKO modelleri aracılığıyla uygulanmaktadır. Hem doğrusallaştırılmış sistem modelinin hem de tüm kapalı çevrim sistemin gürbüzlük özellikleri için, sırasıyla doğrusallaştırılmış doğrusal olmayan sistemin tanılama hatası ve izleme hatası olarak tanımlanan ε-duyarsızlık kayıp fonksiyonu $\ell_{1,\varepsilon}(\cdot,\cdot)$ çalıştırılmaktadır. Sonuçta, önerilen ÖGD tabanlı kararlı, uyarlanır kontrolör DTS modeline ve fiziksel sistemine uygulanmaktadır. Önerilen kontrolörün performansı Oransal-Türev kontrolörle izleme hatası için ortalama karesel hata cinsinden karşılaştırılır.

1. INTRODUCTION

The purpose of the controller design is to find out the appropriate control signal providing the desired behaves for the controlled plant possessing even inherently nonlinearity in a physical system. Control techniques of nonlinear systems has still been a very attractive research field in control systems [1-6]. One of them is a conventional method defined as a linearization method at an equilibrium point around for a nonlinear plant. However, this method cannot cope with the nonlinear system having more than one equilibrium point. Therefore, the feedback linearization method is a powerful technique providing a linear state model which is appropriate for all possible equilibrium points of the nonlinear system [1-11]. In 1989, the feedback linearization method was used on adaptive control of minimum phase systems which can be fully input-output linearized with state feedback [4]. A developed feedback linearization control technique was applied to underactuated mechanical systems such as underactuated robots where some parts of the nonlinear dynamics can be feedback linearized under a condition which is called as strong inertial coupling [12]. Doyle [13] showed that a nonlinear system might be transformed into an equivalent subsystem via input-output linearization by adding a state-dependent constraint to the control input of the subsystem [13]. In 2002, Fuh et al. [14] proposed a method about a feedback linearization of the discrete-time chaotic systems. In 2004, a type of nonlinear systems having time delay systems was addressed the input-output linearization problem solved by a compensator having state and output feedbacks [15]. Ho et al. [16] proposed a feedback linearization-based controller design for a nonlinear benchmark system known as ball and wheel. Herein, the full state feedback is used for transforming from nonlinear system to linear time invariant system with diffeomorphism conditions. Owing to this transformation, linear control techniques might be used for the feedback linearized system. Similarly, in 2010, Zhou et al. [17] developed a feedback linearization-based controller for a quadrotor model having both inner and outer loops. These loops are considered for control design of the attitude and the trajectory tracking of the quadrotor. For a single flexible arm which can be moved against gravity, a feedback linearization-based controller is obtained. Controller has a double loop cascade form. Inner loop includes a controller for better tracking trajectories of motor and cancellation of coulomb friction. Outer loop includes, a linearized model which is formed by input-state linearization [18]. The dynamic model of a wheeled pendulum was analyzed for the controllability and the feedback linearization conditions. The controllability of the system and maximum relative degree are studied and partial feedback linearization is obtained [19,20]. Likewise, a partial feedback linearization technique-based controller was designed for a cart based inverted pendulum [21]. Türker et al. [22] proposed a Lyapunov's direct technique for the stabilization of the inverted pendulum called as Furuta inverted pendulum. Another study on Furuta one, a new trajectory tracking controller was designed via the input-output feedback linearization technique and provided uniformly ultimate bounded error term [23].

In the literature, several studies of artificial neural network (ANN) based feedback linearization have been reported for the nonlinear systems such as nonlinear systems, and robot manipulators [25-33]. ANN is widely used for the feedback linearization technique because of learning behaviours of nonlinear dynamical systems and its generalization ability [24]. He and Unbehauen [28] developed a nonlinear state transformation providing an approximate feedback linearization conditions with a local diffeomorphism implemented via multilayer perceptron (MLP). Likewise, ANN based feedback linearization was proposed as a neuro-controller including fully or partially input–output linearization according to relative degree [29]. ANN based feedback linearization implementations were defined with two nonlinear functions as $f(\circ)$ and $g(\circ)$ implemented by using two separate ANN blocks and the control input form was linearized for the feedback linearized system [24,26,30-33]. The two ANN blocks constitute the feedback linearization implementation so-called the ANN based feedback linearization controllers might be represented as a nonlinear auto regressive moving average (NARMA) model in terms of system input and output [30,31]. Some controllers of feedback linearization techniques were presented as NARMA based neuro controllers [26,32,33]. Şahin [25] proposed a learning feedback linearization method implemented via NARMA model with only one ANN block while the previous studies used two ANN blocks for $f(\circ)$ and $g(\circ)$ nonlinear functions [24,26, 30-33]. Direct adaptive neural controller design was proposed for feedback linearization based nonlinear multi-input multi-output systems [27].

As for stable robust adaptive controller design, it is still a hot topic study area because it produces efficient solutions for nonlinear plants [34-41]. In 2016, the proposed adaptive controller presented as an online controller type for linear time-varying systems and a nonlinear system. Its algorithm was based on a data-dependent Auto-Regressive Moving-Average (ARMA) models for both the controller and the plant. The ARMA models were learned in a supervised learning way with data measured from input-output data pairs of both the plant model and the closed loop system. This NARMA based online robust adaptive controller design is defined as a system identification problem of a partially known the closed loop system. The data dependent adaptive controller parameters were found by minimizing the tracking error for the closed loop system. The stability of the closed loop system was provided by meeting the Schur stability criterion known as a method of solving Diophantine equation called also as Aryabhatta Equation or Bezout identity [34].

This thesis presents an LFL based stable robust adaptive controller design by supervised learning from the data from a plant. The proposed controller is an extension method by exploiting the studies in [24,41] developed data dependent ARMA controller design ensuring the Schur stability conditions for the overall closed-loop system. The proposed algorithm is achieved as follows; i) NARMA based LFL is used to obtain a feedback linearized nonlinear plant by using the ANN, ii) the NARMA-LFL based plant is identified as an ARMA plant model, and iii) the closed-loop control system having ARMA plant and controller models providing Schur stability of it. The training phase of ANN of NARMA based LFL is carried out with a supervised online learning way via both input-output and admissible corresponding states data of the nonlinear plant. Once the training phase of ANN is completed, the feedback linearized nonlinear plant might be defined as the ARMA model including the combination of the nonlinear plant and the LFL block. The proposed stable robust adaptive control algorithm is implemented via the ARMA models of both the plant and the controller. Robustness properties of both the linearized plant model and the overall closed-loop system are employed with the ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ defined as the identification error of the linearized nonlinear plant and the tracking error, respectively [42,43]. Moreover, Schur stability imposed on the overall closed-loop system is guaranteed to determine the linear controller parameters by the linear inequality constraints of the minimization of the $\ell_{1,\varepsilon}(\cdot,\cdot)$ tracking error [41]. The developed adaptive LFL based NARMA controller algorithm is tested on a simulated rotary inverted pendulum (ROTPEN) model and a physical ROTPEN experimental setup. The performance of the proposed adaptive controller is compared with Proportional Derivative (PD) [44,45] controllers. According to the simulation and experimental results, the LFL based NARMA controller performances shows better performances than the other controllers in terms of the Mean Square Error (MSE) for tracking and settling time. Moreover, the proposed adaptive controller based on LFL are analyzed in terms of the ε-insensitiveness effects with MSE under with and w/o noise.

This thesis is organized as follows. In Chapter 2, background on feedback linearization, system modelling and adaptive control. In Chapter 3, the proposed stable robust adaptive NARMA based LFL controller is explained. In Chapter 4, the simulation and experimental results are given. In Chapter 5, conclusion and future direction are presented.

2. BACKGROUND

In this chapter, a background on feedback linearization, system modeling, ARMA and NARMA models, ANN based controllers (with inverse system approximation) and online learning controllers are introduced briefly.

2.1 Feedback Linearization

Let's define the discrete time nonlinear system as in Equation 2.1.

$$\mathbf{x}^{k+1} = f(\mathbf{x}^k, u^k);\ f(\circ): \mathbf{R}^{\mathbf{n}\mathrm{x}1} \rightarrow \mathbf{R}^{\mathbf{n}} \text{ and } u^k \in \mathbf{R} \qquad (2.1)$$

As a nonlinear method, input-state feedback linearization takes place in the related literature because that this type linearization transforms state equations to controllable canonical form [46]. Conventional linearization which includes Taylor expansion about the balance point or balance points is compared with this type linearization in Figure 2.1. Herein, A_Z and b_z are defined as a controllable canonic structure. Conventional linearization around equilibrium points is combined with input-state feedback linearization [25]. A given system can have multiple equilibrium points, multiple local linear state models are obtained from the conventional linearization. A general linear state model is exposed by a feedback linearization and a nonlinear system linearization is applied all equilibrium points [3].

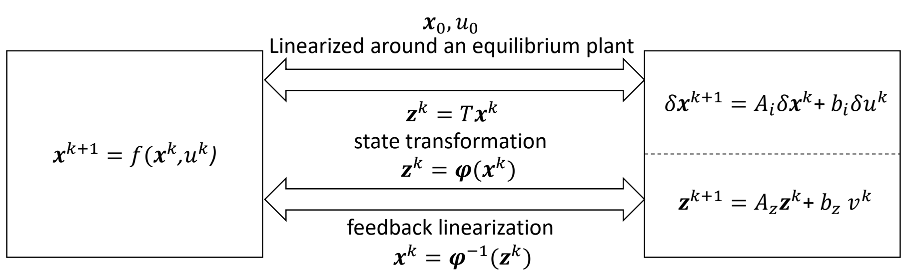


Figure 2.1. Difference between conventional linearization and input-state linearization by feedback

Assuming a discrete single-input single output system (SISO) system given in a form as $\boldsymbol{x}^{k+1} = f(\mathbf{x}^k) + g(\mathbf{x}^k)u^k$, the input-state feedback linearization steps are given below:

- The $g(\mathbf{x}^k) \neq 0$ should be satisfied and the nonlinear system should be in Brunovsky form given in Equation 2.2.
- Nonlinear transformation of states $\mathbf{z}^k = \varphi(\mathbf{x}^k)$ forms the state feedback control law as $u^k = \alpha(\mathbf{x}^k) + \beta(\mathbf{x}^k)v^k = \frac{1}{g(\mathbf{x}^k)}[-f(\mathbf{x}^k) + v^k]$ in Equation 2.2.
- The new input can be defined as v^k.

Considering the above steps are taken place, transformation might be named as input-state linearization given in [46].

$$\begin{bmatrix} x_1^{k+1} \\ . \\ . \\ . \\ x_{n-1}^{k+1} \\ x_n^{k+1} \end{bmatrix} = \begin{bmatrix} x_2^k \\ . \\ . \\ . \\ x_n^k \\ f(\mathrm{x}^\mathrm{k}) + g(\mathrm{x}^\mathrm{k})u^k \end{bmatrix} \tag{2.2}$$

where $f(\circ)$, $g(\circ) : \boldsymbol{R^n} \rightarrow \boldsymbol{R}$ and $u(\circ) : \boldsymbol{R} \rightarrow \boldsymbol{R}$ are defined. The static state feedback controller might be defined as $v^k = \boldsymbol{\Gamma} \boldsymbol{z}^k$ with $\boldsymbol{\Gamma} = [\Gamma_1\ \Gamma_2\ ...\ \Gamma_n]$ defined as linear controller parameters in the linear feedback control loop after completing the feedback linearization steps (Figure 2.2). Therefore, the selection of the appropriate $\boldsymbol{\Gamma}$ might be defined as pole-placement technique for the feedback linearized nonlinear system. Hence, it might be transformed into a linear system which is controllable one in Equation 2.3. The input state linearizable system sufficient conditions are given as follows:

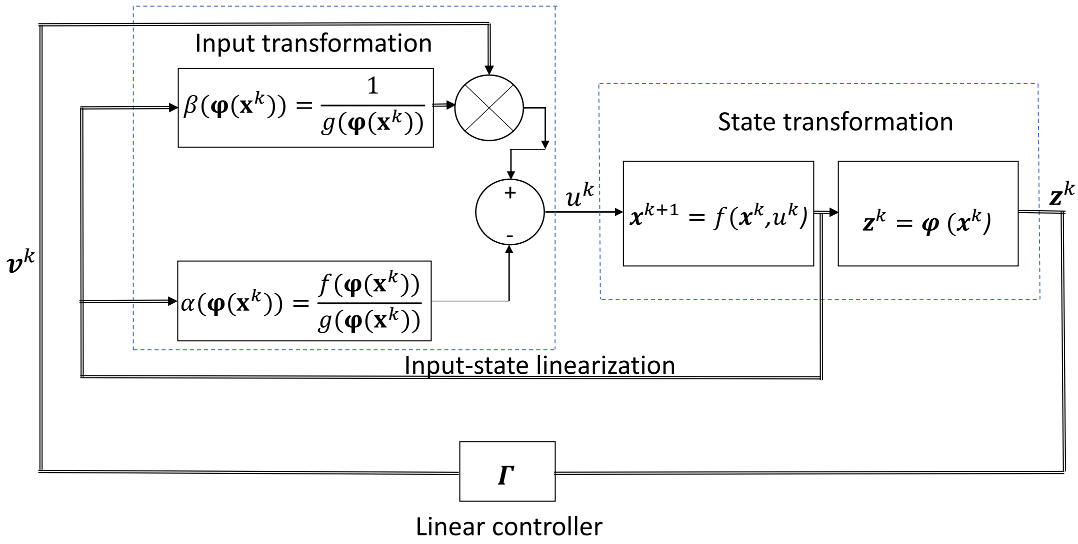


Figure 2.2. Structure about input-output linearized by feedback

Definition 2.1: Assume a SISO system $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$ where $\boldsymbol{f}(\boldsymbol{x}) \in \boldsymbol{R^n}$ and $\boldsymbol{g}(\boldsymbol{x}) \in \boldsymbol{R^n}$ with $\boldsymbol{x} \in \boldsymbol{R^n}$ are smooth vector fields can be said to be input-state linearizable if there exists in a region $\boldsymbol{\Omega} \subseteq \boldsymbol{R^n}$, a C^k diffeomorphic state transformation $\boldsymbol{\varphi}(\circ)\colon \boldsymbol{R^n} \rightarrow \boldsymbol{R^n}$. The nonlinear static state feedback might be defined as $u = \alpha(\boldsymbol{\varphi}(\boldsymbol{x})) + \beta(\boldsymbol{\varphi}(\boldsymbol{x}))v$ and the transformation of the state equations with new state variables $\boldsymbol{z} = \boldsymbol{\varphi}(\boldsymbol{x})$ and the linear control input can be presented as $v = f(\boldsymbol{\varphi}(\boldsymbol{x})) + g(\boldsymbol{\varphi}(\boldsymbol{x}))u = \frac{1}{\beta(\boldsymbol{\varphi}(\boldsymbol{x}))}\left(u - \alpha(\boldsymbol{\varphi}(\boldsymbol{x}))\right)$ which has the linear time-invariant system and internally feedback linearized system in Equation 2.3 [46].

$$\dot{\mathbf{z}} = \begin{bmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ .\;\;.\;\;.\;\;. \\ .\;\;.\;\;.\;\;. \\ .\;\;.\;\;.\;\;. \\ 0\,0\,0\,0 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v = \mathbf{A_Z z} + \mathbf{B_z} v \tag{2.3}$$

Theorem 2.1: With the $\boldsymbol{x} \in \boldsymbol{R^n}$, $\boldsymbol{f}(\boldsymbol{x}) \in \boldsymbol{R^n}$ and $\boldsymbol{g}(\boldsymbol{x}) \in \boldsymbol{R^n}$ are smooth vector fields, the single input nonlinear system is defined as $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$. If and only if a $\boldsymbol{\Omega} \subseteq \boldsymbol{R^n}$ region which provide specified conditions below, system is input-output linearized [46].

- In a $\boldsymbol{\Omega} \subseteq \boldsymbol{R^n}$ region, set of vector fields $\{\boldsymbol{g}, ad_{\boldsymbol{f}}\boldsymbol{g}, \dots, ad_{\boldsymbol{f}}^{n-1}\boldsymbol{g}\}$ are linearly independent.

- Linearly independent vector fields set $\{\boldsymbol{g}, ad_{\boldsymbol{f}}\boldsymbol{g}, \ldots, ad_{\boldsymbol{f}}^{n-2}\boldsymbol{g}\}$ should be involutiveness, that is to say, Lie bracket of any pair of vector fields in the set for vector fields with the linear combinations. The $ad_{\mathbf{f}}\mathbf{g}$ is defined as Lie bracket $[\boldsymbol{f}, \boldsymbol{g}] = (\nabla\boldsymbol{g})\boldsymbol{f} - (\nabla\boldsymbol{f})\boldsymbol{g}$ with gradient operator ∇ w.r.t $\boldsymbol{x^k}$.

The feedback linearized system results of both the discrete nonlinear system and the continuous nonlinear systems with suitable sampling period might be assumed as identically each other. Therefore, the sampled nonlinear continuous-time systems specify the linearized system results [1]. Sufficient conditions of the state transformation of LFL are defined for the local existence via $\boldsymbol{z}^k = \boldsymbol{\varphi}(\boldsymbol{x}^k)$, and a defined control input with a new nonlinear function as $u^k = \Phi(\boldsymbol{z}^k, \boldsymbol{v}^k)$ under $\det\left(\frac{\partial\phi}{\partial\mathbf{v}^k}\right) \neq 0$. Hence, it might be transformed into a linear system which is controllable one in Equation 2.3. The input state linearizable system sufficient conditions are satisfied and it is given in subchapter of LFL.

2.2 System Modelling

The expression of a nonlinear discrete time SISO system is given in Equation 2.4 expanded from Equation 2.1.

$$\boldsymbol{x^{k+1}} = \boldsymbol{f}(\boldsymbol{x^k}, u^k); \quad y^k = \boldsymbol{g}(\boldsymbol{x^k}) \tag{2.4}$$

where $\boldsymbol{f}(\circ)\colon \boldsymbol{R^{nx1}} \rightarrow \boldsymbol{R^n}$, $\boldsymbol{g}(\circ)\colon \boldsymbol{R^{nx1}} \rightarrow \boldsymbol{R}$ and $u^k \in \boldsymbol{R}$. If $\boldsymbol{f}$ and $\boldsymbol{g}$ are not derived from the physical phenomena in a mathematical way, the system modeling can be defined as an identification problem so that the system model is the so-called black box that represents the input-output behavior of the process [47,48].

2.2.1 Blackbox representation

The black box representation can be used as a general approximation for the inputs-outputs of the MIMO system dynamics (Figure 2.3). This representation does not have to be related to the exact model of the considered system; in fact, it is actually focused on the input-output variables of the system. Once the input-output data of the system is obtained, the black-box model can be obtained easily without requiring a clear mathematical knowledge about it [48].

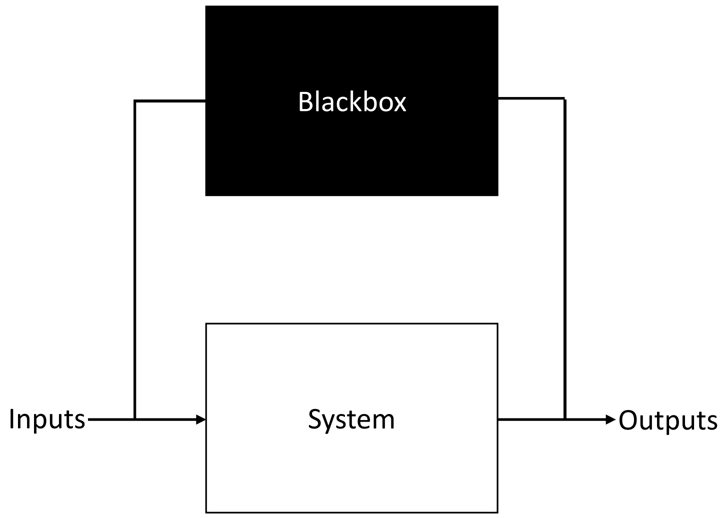


Figure 2.3. A Blackbox structure

2.2.2 ARMA and NARMA models

According to input-output of the considered system, it could be identified with ARMA or NARMA models. A typical ARMA model is given in Equation 2.5 where the first and the second summation parts stand for AR and MA parts [49-51].

$$y(k) = \sum_{i=1}^{N} \alpha_i y(k-i) + \sum_{j=0}^{M} \beta_j u\,(k-j) \tag{2.5}$$

where $\alpha_i \in R$ and $\beta_j \in R$ stand for linear weights, N and M stand for the degrees of AR and MA parts, respectively. ARMA and NARMA models are used as plant models and as controller models in control systems identification and in control system design, respectively. These models employed in a plenty of time series analysis area such as signal processing, image processing, speech recognition, weather forecast, biomedical signal processing [31,52-58].

A corresponding NARMA model of Equation 2.4 can be transformed to Equation 2.6 in which k current time index, N past outputs and M past inputs with a nonlinear function as $H(\circ)\colon \boldsymbol{R}^{N+M+1} \to \boldsymbol{R}$. The NARMA model of the SISO system given in Equation 2.6 can be represented as other forms given in Equation 2.7 and 2.8. These NARMA models having nonlinearities might be implemented by using ANN as an approximator.

$$y(k) = H[y(k-1), y(k-2), \dots, y(k-N); u(k), \dots, u(k-M)] \tag{2.6}$$

$$y(k) = F\left[\sum_{i=1}^{N} \alpha_i y(k-i)\right] + \sum_{j=0}^{M} \beta_j u\,(k-j) \qquad (2.7)$$

$$y(k) = F[y(k-1), \dots, y(k-N)] + G[y(k-1), \dots, y(k-N)]u(k) \qquad (2.8)$$

where $G(\circ): \boldsymbol{R}^{M+1} \to \boldsymbol{R}$ and $F(\circ): \boldsymbol{R}^{N} \to \boldsymbol{R}$.

2.2.3 Artificial neural networks

ANN has been used in control systems area since 1980s because ANN defining a nonlinear algebraic function overcome the nonlinearities and complexity of the control systems. ANN can be defined as a function approximators for any continuous function in a compact set [59]. ANNs have several abilities such as generalization, learning and paralleling and they are used for fault tolerant, supervised and unsupervised learning and optimization. As in the control systems related ANN literature, system identification and controller design are generally achieved by using Multi-Layer Perceptron (MLP) with efficient learning algorithms [60-65]. MLP possesses algebraic neural networks, multi-input and single-output with a sigmoidal activation function (Figure 2.4.).

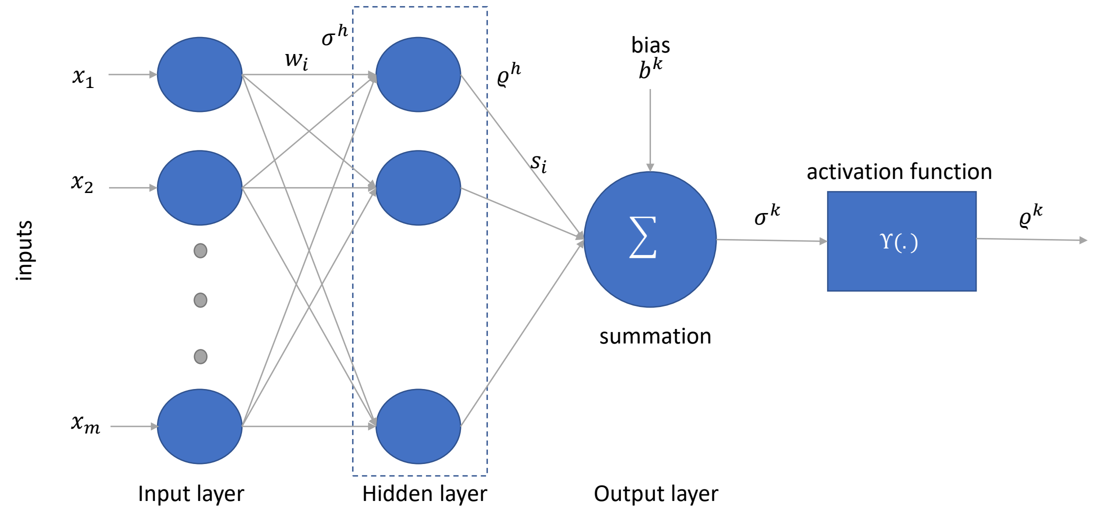


Figure 2.4. An architecture of a typical ANN

A feature of ANN, which learns from the environment and increases its performance during learning phase, updates weights and bias values between neurons. At the end of each iteration, the system receives more information from the environment and improves system performance. MLP determines connection weights of the neurons connections with error back-propagation (BP) algorithm which is based on a gradient descent technique. It finds generally local minimum of the squared error in Equation 2.9 between the desired and actual outputs. The partial derivatives of the output error are calculated by BP where partial derivatives are found with respect to connection weights (Equation 2.10 and 2.11).

$$E = 1/2(r - \varrho)^2 \tag{2.9}$$

$$\frac{\partial E}{\partial s_i} = \frac{\partial E}{\partial \varrho}\frac{\partial \varrho}{\partial \sigma}\frac{\partial \sigma}{\partial s_i} = -(r - \varrho)\Upsilon'(\sigma)\varrho^h \tag{2.10}$$

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial \varrho}\frac{\partial \varrho}{\partial \sigma}\frac{\partial \sigma}{\partial \varrho^h}\frac{\partial \varrho^h}{\partial \sigma^h}\frac{\partial \sigma^h}{\partial w_i} = -(r - \varrho)\Upsilon'(\sigma)s\Upsilon'(\sigma^h)x \tag{2.11}$$

where the derivative of the sigmoidal nonlinearity is denoted $\Upsilon'(\circ)$ found as a sigmoidal function. In the opposite of the gradient direction, to update the connection weights (Equation 2.13), using a step size ζ which is sufficiently small and called as learning rate.

$$s(k+1) = s(k) - \zeta\,\frac{\partial E}{\partial s(k)} = s(k) + \zeta(r - y)\Upsilon^{'}(\sigma)\varrho^h \tag{2.12}$$

$$w_i(k+1) = w_i(k) - \zeta\,\frac{\partial E}{\partial w(k)} = w(k) + \zeta(r - \varrho)\Upsilon^{'}(\sigma)s\Upsilon^{'}\left(\sigma^h\right)x \tag{2.13}$$

2.3 ANN Based System Identification

The ability to approximate the nonlinear functions of the ANN allows for the use in system identification issues. There are two types of identification structure with ANN as parallel and series-parallel. In parallel mode, it is designed via system inputs and model outputs providing an ARMA model in the Equation 2.14 (Figure 2.5). In series-parallel mode, ANN based identification is formed via inputs-outputs of the system providing an ARMA model in the Equation 2.15 (Figure 2.6).

$$\hat{y}(k) = \sum_{i=1}^{N} \alpha_i \hat{y}(k-i) + \sum_{j=0}^{M} \beta_j u\,(k-j) \tag{2.14}$$

$$\hat{y}(k) = \sum_{i=1}^{N} \alpha_i y(k-i) + \sum_{j=0}^{M} \beta_j u\,(k-j) \tag{2.15}$$

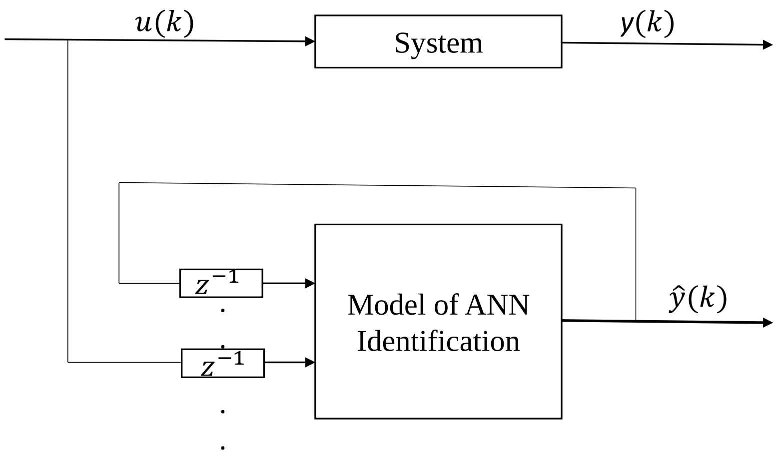


Figure 2.5. Model of ANN based parallel identification

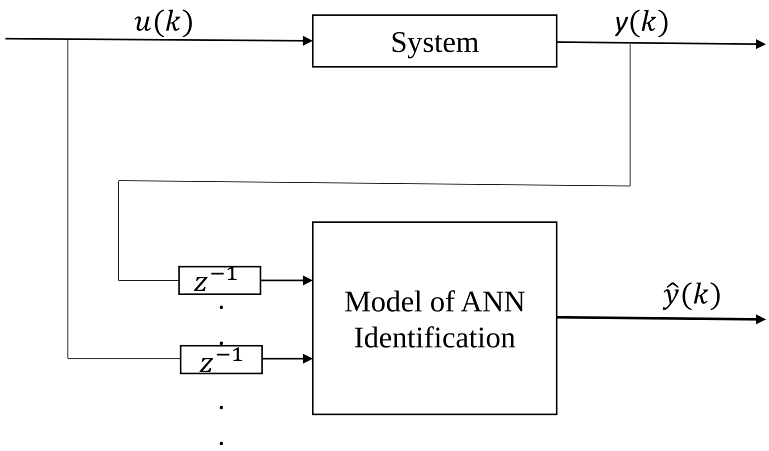


Figure 2.6. Model of ANN based series-parallel identification

2.4 ANN Based Controllers

Werbos and Narendra did firstly report ANN based controllers via their learning capabilities, coping with nonlinearity, and their reactions to parameter changes [66, 67]. ANN based controllers can be divided to two groups as follows: i) the feedforward ANN which is also called as algebraic ANN, and ii) the recurrent ANN which is also called as dynamical ANN [67,68]. As for the controller design strategies, the first strategy is that direct inverse control method provides identity system via mapping from the reference signal $r(k)$ to actual plant output $y(k)$. Herein, ANN is trained for inverse system and used as a controller (Figure 2.7).

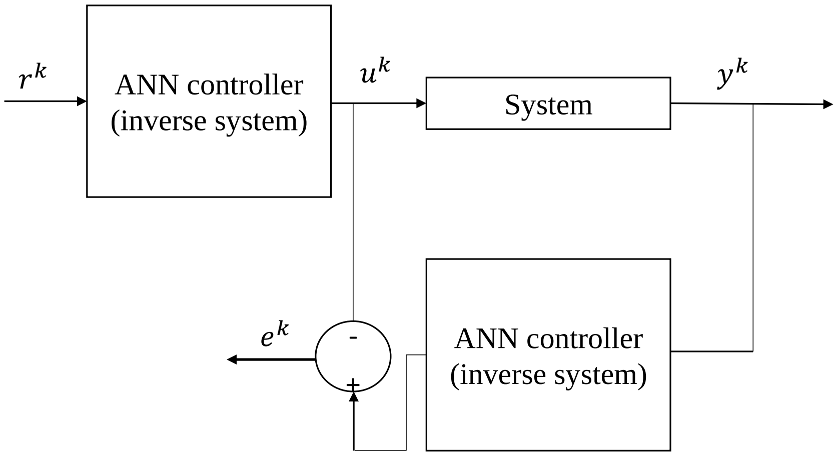


Figure 2.7. ANN based direct inverse control strategy

Likewise, the feed-forward inverse controller strategy having two different ANN blocks and trained with two phases. In the first phase, ANN of the system identification is completed with the control and the system output signals. The second phase, ANN based controller (i.e. inverse system block) is trained by using ANN based system identification block according to the closed loop system error minimization (Figure 2.8).

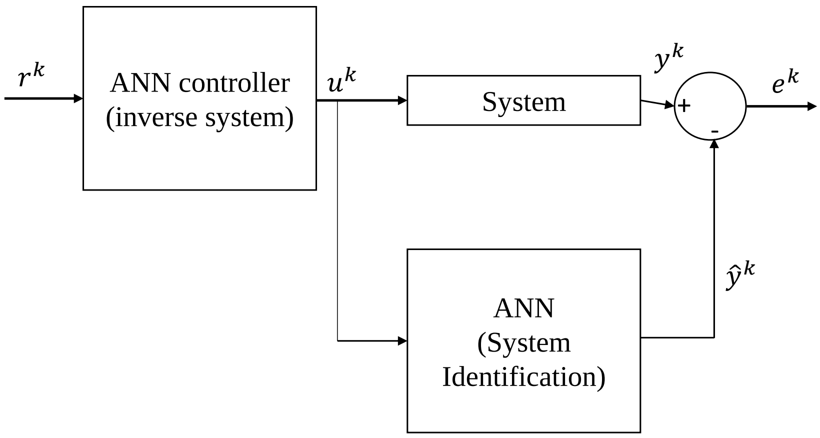


Figure 2.8. A structure of feed-forward inverse control

2.5 Adaptive Control Methods

The adaptive control methods are powerful algorithms for overcoming system uncertainties, its parameters changing and disturbances problems. The control parameters are updated in each iteration step in terms of online control applications [46,70-76]. One of the adaptive control method is self-tuning regulator (STR) finding out the plant parameters via the stochastic estimation in an online way in simultaneously updating the controller parameters given in Figure 2.9 [34,46].

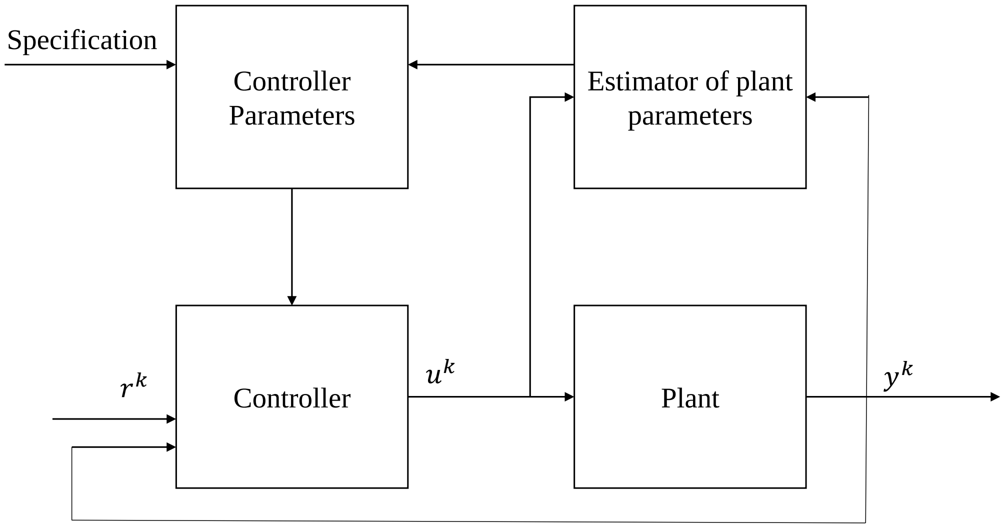


Figure 2.9. STR controller method

The other adaptive control method is well known structure is model reference adaptive control (MRAC) method having a stable reference model. This controller can eliminate disturbances, parameter variations, and system uncertainties given in Figure 2.10 [25, 46,77,78].

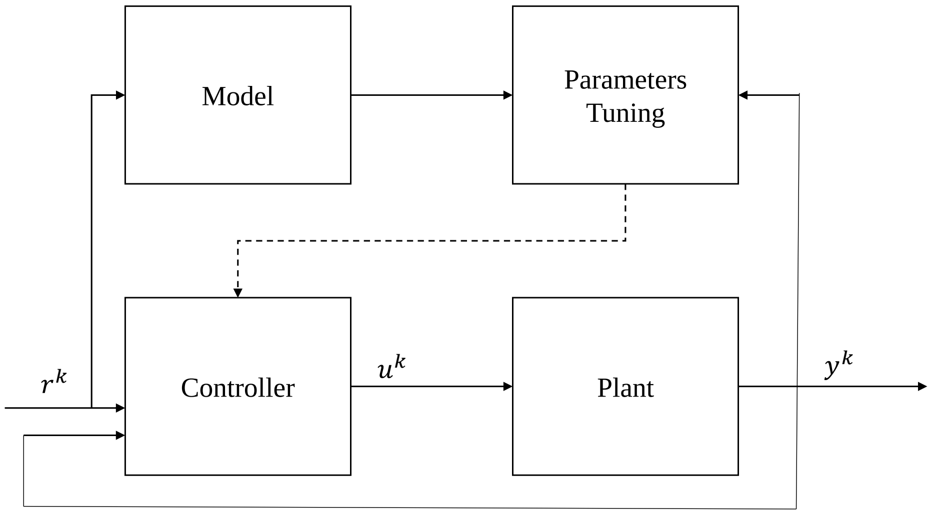


Figure 2.10. MRAC method

2.6 Adaptive Controller Design of Partially Known Closed Loop System

Partially known closed-loop control system has an identified or a known plant and the adaptive controller design unknown part in online mode. Plant is firstly identified and then the adaptive controller parameters are determined via tracking error minimization of the closed loop system [41]. It can be defined a combined method by using STR and MRAC. The adaptive controller structure which has two degrees of freedom is depicted in Figure 2.11. The algorithms of the plant identification and the adaptive controller are trained and updated in simultaneously by a supervised learning way.

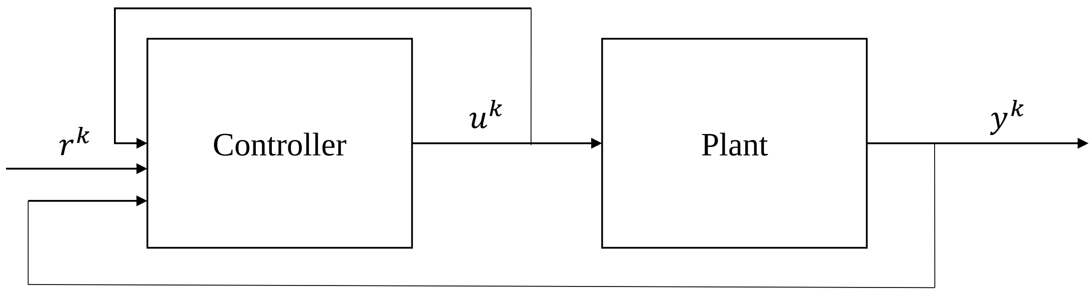


Figure 2.11. A control structure which has two degrees of freedom

2.7 Closed Loop Stability and Robustness

Schur stability criteria is related to absolute stability of the discrete time systems. If the roots of the system characteristic equation are in the unit disk, the system might be bounded-input bounded-output (BIBO) stable for linear time invariant systems. There exists a characteristic polynomial under it has not pole-zero cancellation, the polynomial might be defined as $p(z) = \mu_n z^n \cdots + \mu_2 z^2 + \mu_1 z + \mu_0$ where $\mu_i (i = 0, 1, \ldots, n)$ are the real numbers. Sufficient stability conditions of that system might be defined with linear inequality constraints as follows $\mu_n > \cdots > \mu_1 > \mu_0 > 0$ s for the Schur stability [79].

As for robustness issue of the closed-loop system, $\varepsilon -$ insensitive loss functions might be used for robustness property given in detail in the following Chapter 3.

$$M(y - \hat{y}) = \ell_1(|y - \hat{y}|_\varepsilon) \quad (2.19)$$

$$|y - \hat{y}|_\varepsilon = \begin{cases} 0\ , & if(|y - \hat{y}| \le \varepsilon \\ |y - \hat{y}| - \varepsilon & otherwise \end{cases}$$

The loss function might be described as absolute norm representing as ℓ_1 norm. The loss is equal to 0 if the discrepancy between the predicted and the observed values is less than ε (Figure 2.12) [80].

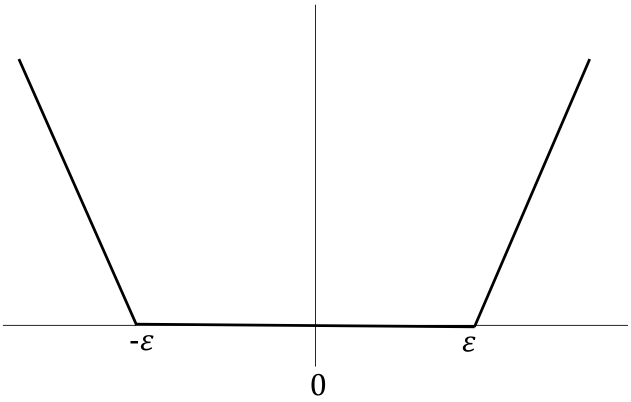


Figure 2.12. $\varepsilon -$ insensitive ℓ_1 based loss function

3. PROPOSED LFL BASED ADAPTIVE CONTROLLER

The proposed adaptive controller algorithm is based on a feedback linearized plant model with LFL obtained via ANN. The proposed stable robust adaptive algorithm is achieved by three progressive stages as follows (Figure 3.1); I) NARMA based LFL strategy is used to obtain a feedback linearized nonlinear plant by using ANN, ii) the NARMA-LFL based feedback linearized plant might be identified as an ARMA plant model with ε-insensitive loss function for system identification, and iii) the overall closed-loop control system providing Schur stability conditions and ε-insensitive loss function for tracking error is constituted by both ARMA plant and controller model. All three stages are shown in Figure 3.2.

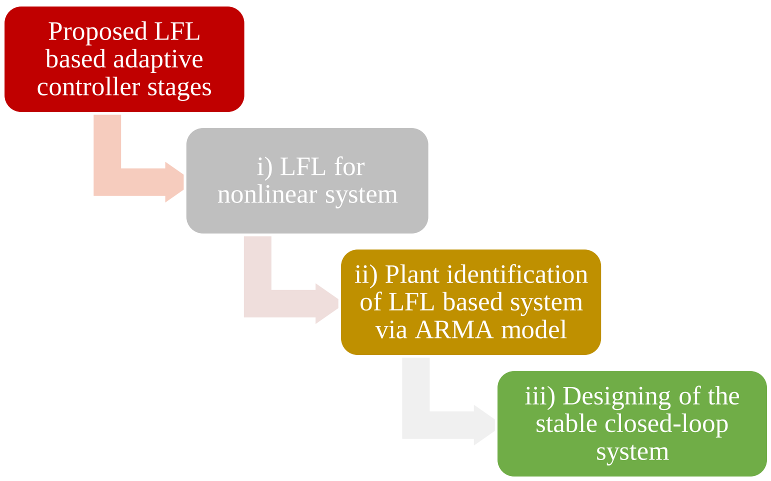


Figure 3.1. Stages of the proposed adaptive controller

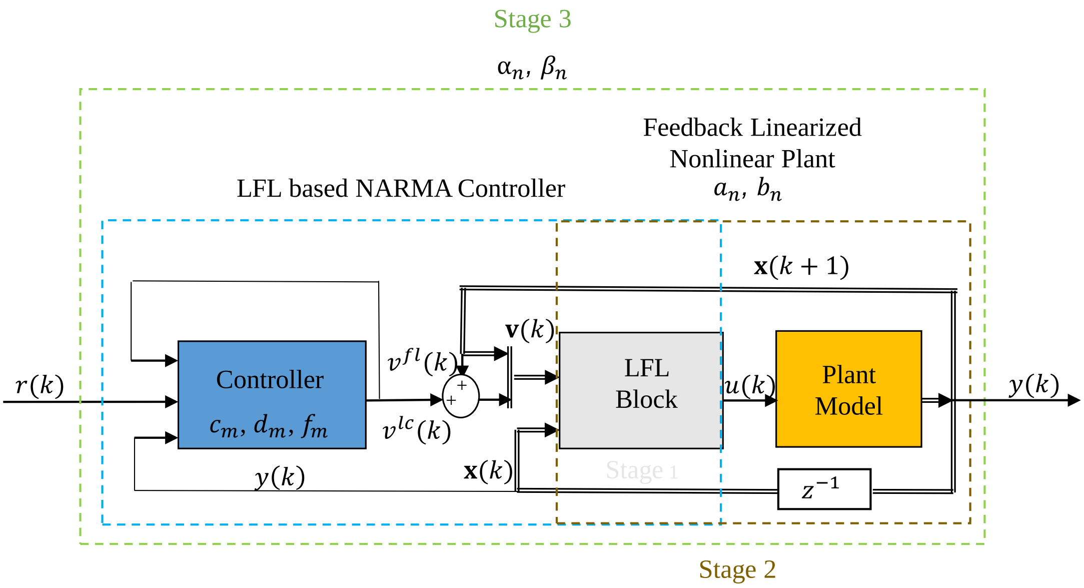


Figure 3.2. The proposed LFL based stable adaptive controller

3.1 LFL For Nonlinear Systems

Although linear systems don't need to be the feedback linearized inherently, for a better understanding of the LFL strategy is briefly explained with a linear SISO case in this Subchapter. So, first of all, let's define a discrete time linear system as in Equation 3.1 where $\boldsymbol{A} \in \boldsymbol{R}^{nxn}$, $\boldsymbol{B} \in \boldsymbol{R}^{nxm}$ and $u^k \in \boldsymbol{R}$ is control input while $\boldsymbol{x}^k \in \boldsymbol{R}^n$ is defined as state vector.

$$\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{B}u^k \tag{3.1}$$

Assuming that $\boldsymbol{x}^{k+1} \coloneqq \boldsymbol{v}^k$ in which $\boldsymbol{v}^k$ stands for the feedback linearized input of the system obtained from the states of the system in Equation 3.1 might be transformed to Equation 3.2.

$$f(\boldsymbol{x}^k, u^k) \coloneqq \boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{B}u^k \tag{3.2}$$

where $f(\boldsymbol{x}^k, u^k)$ is a vector field with $f(\circ) : \boldsymbol{R}^{nxm}$. The control input of the linear system is rewritten as $u^k = \boldsymbol{B}^{-1}[\boldsymbol{v}^k - \boldsymbol{A}\boldsymbol{x}^k]$, if and only if $\boldsymbol{B}^{-1}$exists.

Likewise, as far as the nonlinear system case concerned, a discrete time nonlinear system is defined as in Equation 3.3.

$$\boldsymbol{x}^{k+1} = \boldsymbol{f}(\boldsymbol{x}^k, u^k) \tag{3.3}$$

where $\boldsymbol{f}(\circ)$: $\boldsymbol{R}^{nxm} \rightarrow \boldsymbol{R}^n, \boldsymbol{x}^{\boldsymbol{k}} \in \boldsymbol{R}^{\boldsymbol{n}}$ is defined as state vector, and $u^k \in \boldsymbol{R}$ is the control input. Assuming that the feedback linearized input vector is considered as $\mathbf{x}^{\mathrm{k+1}} \coloneqq \mathbf{v}^{\mathrm{k}}$, Equation 3.3 might be transformed to Equation 3.4.

$$\boldsymbol{f}(\boldsymbol{x}^k, u^k) \coloneqq \boldsymbol{v}^k \tag{3.4}$$

The control input of the nonlinear system might be written as Equation 3.5 represented a nonlinear function $\Phi(\circ)$: $\boldsymbol{R}^{nxm} \rightarrow \boldsymbol{R}$ with states and feedback linearized inputs of the nonlinear system.

$$u^k = \Phi(\boldsymbol{x}^k, \boldsymbol{v}^k) \tag{3.5}$$

The nonlinear system is assumed as $\mathbf{x}^{k+1} = f(\mathbf{x}^k) + g(\mathbf{x}^k)u^k$ as a type of a general nonlinear system representation of Equation 3.4. The control input of the nonlinear system is obtained in the following form as $u^k = \frac{1}{g(\mathbf{x}^k)}[v^k - f(\mathbf{x}^k)]$ if and only if $g(\mathbf{x}^k) \neq 0$.

As for implementation of LFL based algorithm with ANN, the LFL block might be formed with a suitable MLP-ANN possessing one hidden layer (Figure 3.3) where MLP based LFL block is trained with u^k and $\{(\boldsymbol{x}^k, \boldsymbol{v}^k)\}_{k=0}^{\overline{\overline{K}}}$ as both output and inputs, respectively, and $\overline{\overline{K}}$ denotes finite natural number. According to desired output u^k of the LFL block, the LFL training error is tried to minimize at each data sample $\boldsymbol{k}$.

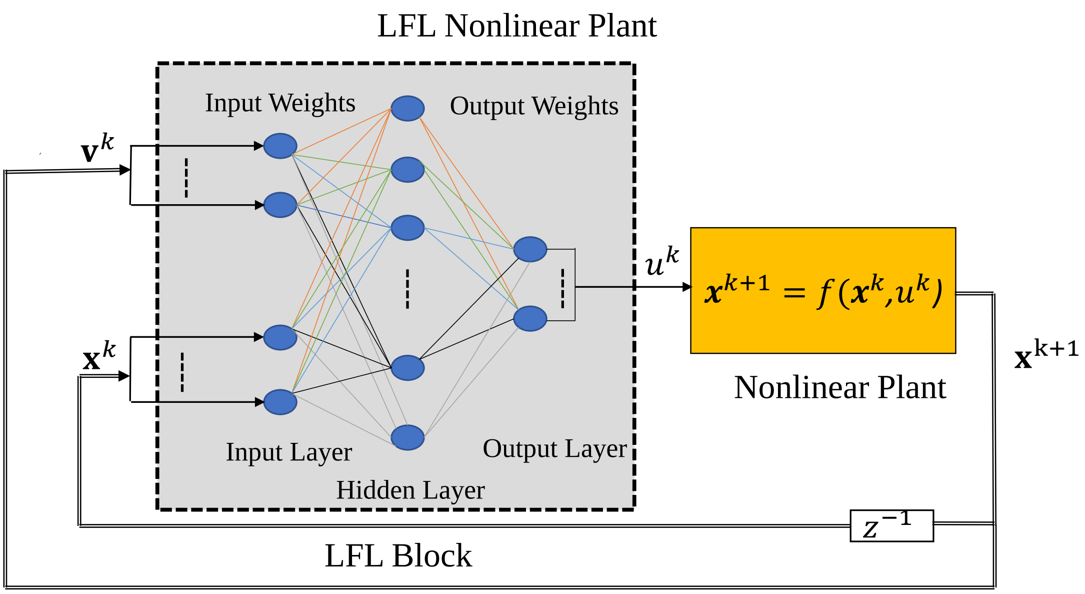


Figure 3.3. MLP-ANN Based LFL Block

3.2 Plant Identification of the LFL Based System via ARMA Model

Let's define an ARMA model to be identified system model given as in Equation 3.6 for a SISO nonlinear system.

$$y(k) = \sum_{n=1}^{N} a_n y_a(k-n) + \sum_{n=0}^{M} b_n\, v^{lc}(k-n) \tag{3.6}$$

where a_n and b_n stands for model parameters of the identified LFL based plant, $y_a(k)$ is the plant output and $v^{lc}(k)$ is the linear controller output of the closed loop system called as the one input of LFL nonlinear system. The plant identification of the LFL based nonlinear plant system is depicted as block diagram in Figure 3.4 by using ARMA model.

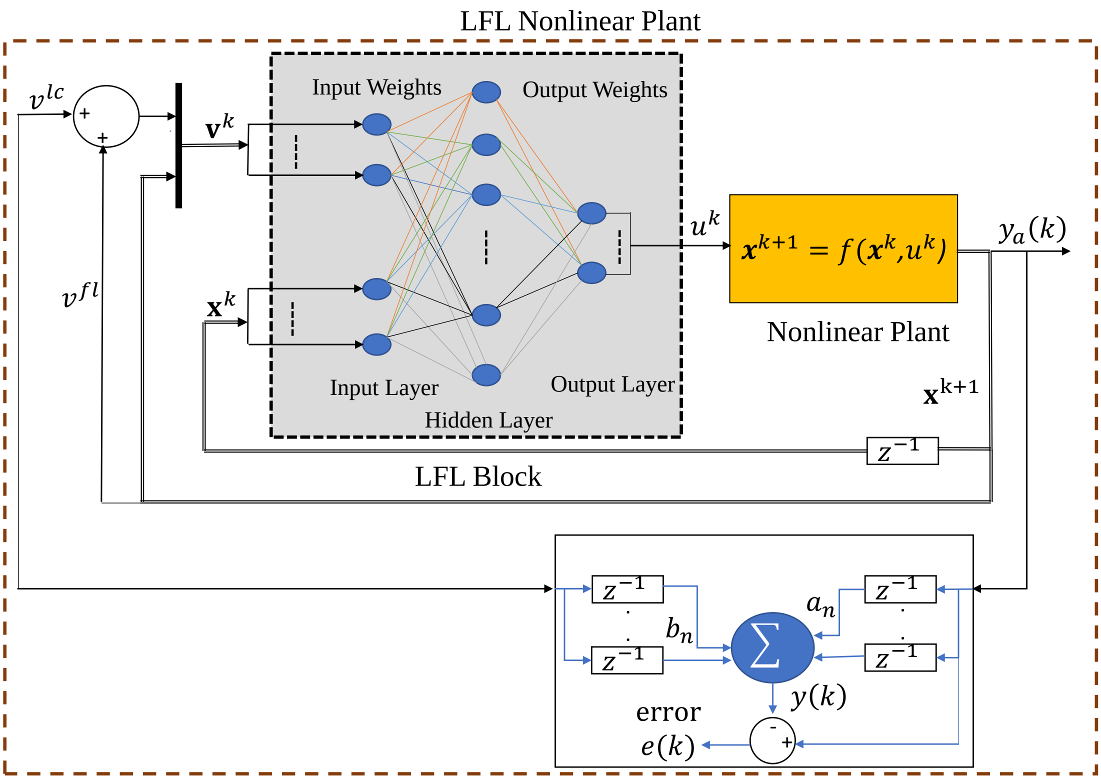


Figure 3.4. ARMA plant identification of the LFL based nonlinear plant

The system identification with ARMA modeling is achieved by minimizing the identification error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 3.7 in terms of the time interval of $[k, k-K+1]$ in an offline manner. Herein, the K is sliding window length for ARMA plant identification.

$$\frac{1}{K}\sum_{s=0}^{K-1} \ell_{1,\varepsilon}\begin{pmatrix} y_a(k-s), \sum_{n=1}^{N} a_n y(k-s-n) \\ +\sum_{n=0}^{M} b_n v^{lc}(k-s-n) \end{pmatrix} + \lambda \left\| \begin{matrix} a \\ b \end{matrix} \right\|_2^2 \qquad (3.7)$$

where ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ in Equation 3.7 is a measurement of the distance between the $(k-s)^{\text{th}}$ actual output sample $y_a(s)$ of plant and the $(k-s)^{\text{th}}$ output sample $y(k-s) = \sum_{n=1}^{N} a_n y(k-s-n) + \sum_{n=0}^{M} b_n v^{lc}(k-s-n)$ of plant model. Absolute norm ε-insensitive loss function can be defined as $\ell_{1,\varepsilon}(y_a(s), y(s)) = |y_a(s) - y(s)|$ if $|y_a(s) - y(s)| \geq \varepsilon$ and $\ell_{1,\varepsilon}(y_a(s), y(s)) = 0$ if $|y_a(s) - y(s)| < \varepsilon$ [80]. Herein, $\varepsilon-$insensitiveness is represented for having robustness against measurement noise, disturbances, and small variations in the output of the plant. $\left\|\begin{matrix} a \\ b \end{matrix}\right\|_2^2 := \sum_{n=1}^{N} a_n^2 + \sum_{n=0}^{M} b_n^2$, that is to say, the square of the Euclidean norm of the model parameters providing nonzero results, λ is the regularization term which provides a smooth model avoiding over-fitting which might be defined as more general model of the plant.

3.3 Designing of The Stable Adaptive Closed-Loop System

This sub-chapter describes the stable adaptive ARMA controller design stages of the closed-loop system. The ARMA controller is considered as a system identification problem in a closed-loop control system having a real/model plant and a controller blocks under unity feedback assumption. Indeed, the proposed adaptive controller design might also be noted as a closed-loop control system identification problem whose parameters are partially known after identification of the plant to be controlled [41]. Hence, after the plant identification stage, the second stage is that the proposed adaptive ARMA controller parameters with the known closed-loop input-output data can be found by solved by optimization techniques with linear constraints in a manner of the supervised learning algorithm (Figure 3.5).

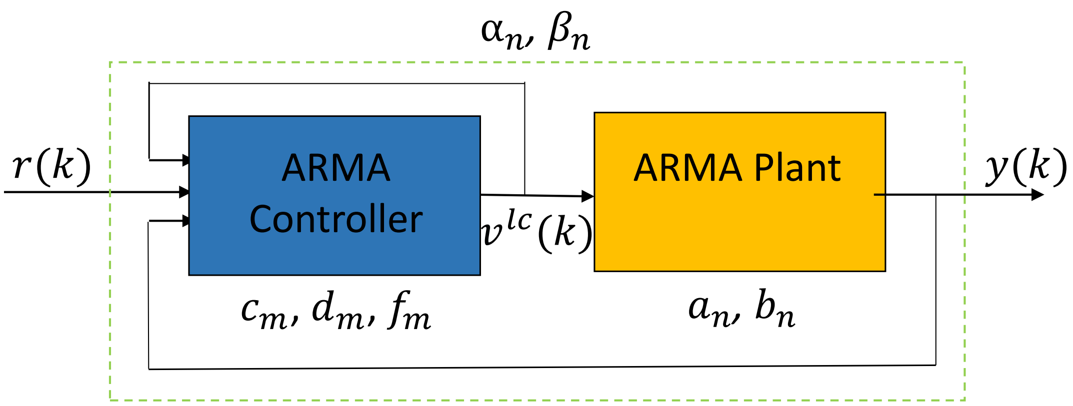


Figure 3.5. ARMA based closed-loop system with two degree of freedom design

The ARMA controller model is defined as a SISO system in terms of $v^{lc}(k)$ and $r(k)$ so that it might be given as a two degree-of-freedom structure in Equations 3.8 [34].

$$v^{lc}(k) = \begin{Bmatrix} \sum_{m=1}^{P} f_m v^{lc}(k-m) + \sum_{m=0}^{R} c_m r(k-m) \\ + \sum_{m=0}^{Q} d_m y(k-m) \end{Bmatrix} \tag{3.8}$$

where $v^{lc}(k)$, $r(k)$ and $y(k)$ stands for the control input, the reference or desired output, and the closed-loop system output. Likewise, c_m, d_m and f_m stands for the adaptive controller parameters to be determined. As for overall the closed-loop system identification, the ARMA model might be found with the α_n, and β_n parameters using the definitions $\widehat{N} =: max\{P+N, M+Q\}$ and $\widehat{M} =: M+R$ in manner of optimization techniques with linear constraints in Equation 3.10. Herein, the algebraic equations might be obtained in Equations 3.7-3.9 solved by using Diophantine equations [34].

$$y(k) = \sum_{n=1}^{\widehat{N}} \alpha_n y(k-n) + \sum_{n=0}^{\widehat{M}} \beta_n \, r(k-n) \tag{3.9}$$

$$a_o \coloneqq 1 + a_0 \, f_0 \text{-} b_0 \, d_0 \tag{3.10}$$

$$a_i \coloneqq \sum_{j=0}^{i} a_j f_{i-j} - \sum_{j=0}^{i} b_j d_{i-j} \text{ for } i \in \{1,2,\dots,N\}$$

$$a_i \coloneqq \sum_{j=i-N}^{N} a_j f_{i-j} - \sum_{j=i-N}^{N} b_j d_{i-j} \text{ for } i \in \{N+1, N+2, \dots, 2N\}$$

$$\beta_i \coloneqq -\sum_{j=0}^{i} b_j c_{i-j} \text{ for } i \in \{0,1,2,\dots,N\}$$

$$\beta_i \coloneqq -\sum_{j=i-N}^{N} b_j c_{i-j} \text{ for } i \in \{N+1, N+2, \dots, 2N\}$$

Measured input-output data set of the system to be controlled and desired output-reference input data set can be written as $\{v^{lc}[k-s,N], y_a[k-s,N]\}_{s=0}^{K-1}$, $\{r[k-s,N], y_d[k-s,N]\}_{s=0}^{L-1}$ respectively, to required fields of Equation 3.6, 3.8 and 3.9 where $y_a(k), r(k)$ and $y_d(k)$ stands for actual output, reference and desired output. In these data sets, the current and previous N samples of any signal are represented as $x[t,N] \coloneqq [x(t), x(t-1), \dots, x(t-N)]$.

The closed-loop system identification with ARMA model is fulfilled by minimizing the tracking error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 3.10 in terms of the time interval of $[k, k-L+1]$ in an offline manner. Herein, the L is sliding window length for ARMA model identification of the closed-loop system.

$$\frac{1}{L}\sum_{s=0}^{L-1} \ell_{1,\varepsilon}\begin{pmatrix} y_d(k-s), \sum_{n=0}^{\widehat{N}} \alpha_n y(k-s-n) \\ + \sum_{n=0}^{\widehat{M}} \beta_n r(k-s-n) \end{pmatrix} + \lambda \left\| \begin{matrix} \alpha \\ \beta \end{matrix} \right\|_2^2 \tag{3.11}$$

where ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ in Equation 3.10 is a measurement of the distance between the $(k-s)^{\text{th}}$ desired output sample $y_d(s)$ of plant and the $(k-s)^{\text{th}}$ output sample $y(k-s) = \sum_{n=0}^{\hat{N}} \alpha_n y(k-s-n) + \sum_{n=0}^{\hat{M}} \beta_n r\,(k-s-n)$ of the closed-loop system model. Absolute norm ε-insensitive loss function might be defined as $\ell_{1,\varepsilon}(y_d(s), y(s)) = |y_d(s) - y(s)|$ if $|y_d(s) - y(s)| \geq \varepsilon$ and $\ell_{1,\varepsilon}(y_d(s), y(s)) = 0$ if $|y_d(s) - y(s)| < \varepsilon$. Herein, $\varepsilon-$insensitiveness is represented for having robustness against measurement noise, disturbances, and small variations in the output of the closed-loop system [80]. $\left\|\begin{matrix}\alpha\\ \beta\end{matrix}\right\|_2^2 := \sum_{n=0}^{\hat{N}} \alpha_n^2 + \sum_{n=0}^{\hat{M}} \beta_n^2$, that is to say, the square of the Euclidean norm of the closed-loop system model parameters providing nonzero results, λ is the regularization term which provides a smooth model avoiding over-fitting which might be defined as more general model of the closed-loop system. Moreover, in order to ensure the stability of the closed loop system, constraints $\propto_0 > \cdots > \propto_{2N-1} > \propto_{2N} > 0$ of Schur stability conditions are applied as linear constraint equations in minimizing tracking error of the closed-loop system given in Equation 3.10 [79].

During the learning phase of the controller and the identification of the system to be controlled, the proposed stable robust adaptive ARMA controller design can be performed in two training modes such as batch and sliding window. In the batch mode, the time interval constituting the entire data set is used and the parameters are not updated over time. However, in sliding window mode, ARMA models parameters of the plant, the closed-loop system and controller are updated in K and L window lengths.

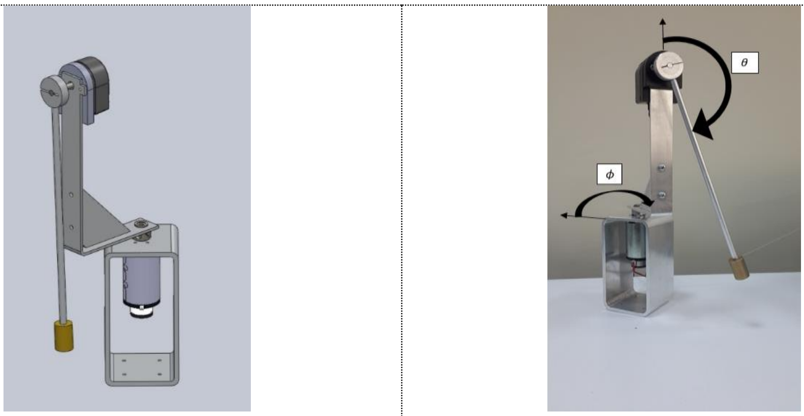
4. SIMULATION AND EXPERIMENTAL RESULTS

In this section, the ROTPEN which is also known as Furuta inverted pendulum model and the physical ROTPEN experimental setup are briefly described. The developed stable robust adaptive LFL based NARMA controller algorithm is tested on the ROTPEN model and its physical experimental setup. The performances of both the proposed adaptive and PD controllers are compared in terms of the settling time and MSE of tracking errors, and ε-insensitiveness effects under with and w/o noise.

4.1 ROTPEN Experimental Setup and Model

The ROTPEN is one of the most popular benchmark experimental setup used in the field of nonlinear control applications. It is also an example of a well-known under-actuated mechanical system [19-23, 44, 81]. Incompletely driven mechanical systems are widely used in the field of robotics, and the main feature of these systems is that they have fewer actuators than degrees of freedom [82]. The inverted pendulum possesses unstable and non-linear dynamical behaviors inherently. Another important feature that makes the rotary inverted pendulum more interesting is that it forms the basis of many new technologies such as seismometers, humanoid robots, unmanned air vehicles and rockets [83]. The ROTPEN system has one input and one output which might be chosen one of two states. The input of the ROTPEN is fed with the force, the output might be selected as either the pendulum angle or the angular position of the base. Therefore, the ROTPEN might be represented as a single-input multiple-output (SIMO) system) [84].

The ROTPEN experimental setup consists of mechanical design, data acquisition card, and software. The mechanical design of the pendulum is made by SolidWorks software and is given in Figure 4.1a. A direct current motor is used to rotate the ROTPEN arm horizontally. The pendulum is connected to the pendulum arm by the pivot. Thus, the pendulum will be able to oscillate easily. AVAGO HEDM-5505-j06 two-channel 1024 resolution encoder is located on the shaft. This encoder was used to measure the angle of the pendulum with the horizontal plane and to implement the control system. The end of the L-shaped pendulum arm is mounted on the shaft of the dc motor. Due to the circular rotation of the motor shaft, the pendulum arm can be moved clockwise and counterclockwise. The angle of the arm is calculated with the encoder mounted on the motor. A rotating arm in a horizontal axis and a rotating pendulum which is mounted on arm, in a vertical plane take part in the rotary inverted pendulum [81]. The final version of the successful ROTPEN setup is given in Figure 4.1b. and Figure 4.2. In the software part, control algorithms designed in MATLAB environment are used in Simulink environment.



(a)	(b)

Figure 4.1. (a) ROTPEN Solidworks design (b) designed ROTPEN setup.

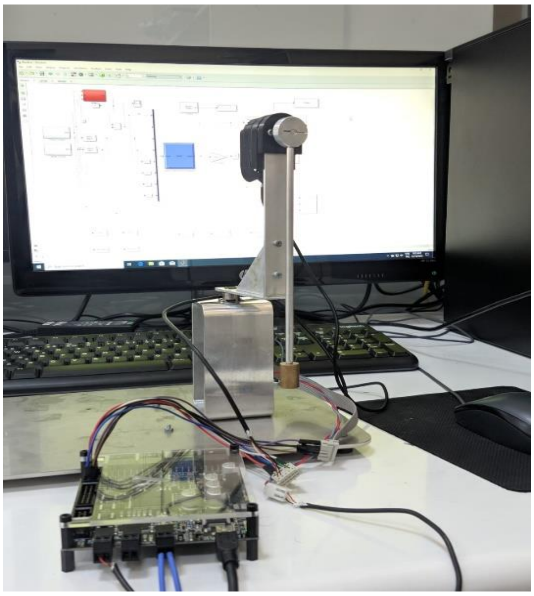


Figure 4.2. V-DAQ data acquisition card and ROTPEN experimental setup.

As for a typical ROTPEN modelling, the variables and parameters of the dynamical system model are given in Figure 4.3. Mathematical derivations results of the total kinetic energy with dynamical system equations of the ROTPEN system are denoted in Equation 4.1 where θ and ϕ stands for the pendulum angle and the rotating arm angle, respectively, $\tau_{output} = \frac{K_t\left(V_m - K_m\left(\frac{d}{dt}\phi(t)\right)\right)}{R_m}$ is used for torque control equation of the dc motor. ROTPEN system parameters are borrowed from [11] (Table 4.1).

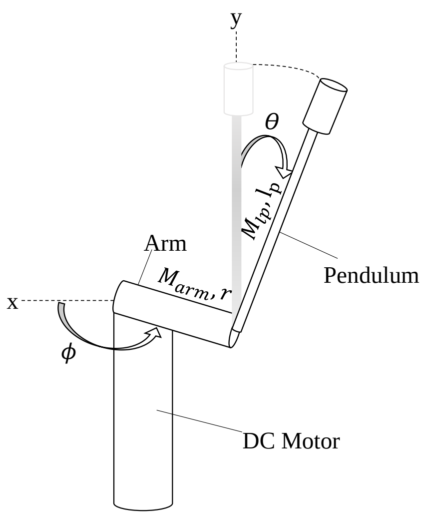


Figure 4.3. ROTPEN solid model with variables.

$$\frac{d^2}{dt^2}\phi(t) = \frac{M_p^2 g l_p^2 r \cos(\phi(t))\theta(t)}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}} - \frac{J_p M_p r^2 \cos(\phi(t))\sin(\phi(t))\left(\frac{d}{dt}\phi(t)\right)^2}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}} - \frac{J_p \tau_{output} + M_p l_p^2 \tau_{output}}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}} \quad (4.1)$$

$$\frac{d^2}{dt^2}\theta(t) = \frac{l_p M_p\left(-J_{eq} g + M_p r^2 \sin(\phi(t))^2 g - M_p r^2 g\right)\theta(t)}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}} - \frac{l_p M_p r \sin(\phi(t)) J_{eq}\left(\frac{d}{dt}\phi(t)\right)^2}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}} + \frac{l_p M_p r \tau_{output} \cos(\phi(t))}{\left(M_p r^2 \sin(\phi(t))^2 - J_{eq} - M_p r^2\right)J_p - M_p l_p^2 J_{eq}}$$

Table 4.1. Descriptions of ROTPEN system parameters and their values.

Symbol	Description (Unit)	Value
K_m	Electromotive torque constant of the motor ($V/(rad/s)$)	0.0333
g	Gravity acceleration ($kg.m^2$)	9.81
B_{eq}	Arm viscous damping ($N.m.s/rad$)	0
B_p	Pendulum viscous damping ($N.m.s/rad$)	0
K_t	Motor torque constant ($N.m$)	0.0333
R_m	Armature resistance of the motor (Ω)	8.7
V_m	Motor input voltage ($Volt$)	0-24
M_{arm}	Mass of rotary arm (kg)	0.08
M_p	Mass of the pendulum “link and weight included” (kg)	0.027
r	Length of rotary arm (m)	0.0826
l_p	Length of inverted pendulum (m)	0.153
J_{eq}	Inertia rotary of rotary arm ($kg.m^2$)	0.000368
J_p	Inertia rotary of inverted pendulum ($kg.m^2$)	0.000698

4.2. The Simulation Results of the Proposed Controller for ROTPEN Model

The proposed adaptive controller is tested on simulated ROTPEN model on a MATLAB environment. The proposed algorithm is achieved by three progressive stages as follows; i) NARMA based LFL strategy is used to obtain a feedback linearized nonlinear plant by using ANN, ii) the NARMA-LFL based feedback linearized plant might be identified as an ARMA plant model with ε-insensitive loss function for system identification, and iii) the overall closed-loop control system providing Schur stability conditions and ε-insensitive loss function for tracking error is constituted by both ARMA plant and controller model. These stages of the simulation studies are explained in the following Subsections.

4.2.1 LFL for ROTPEN model

To achieve NARMA based LFL via MLP for the ROTPEN model, training data set is formed. The data set consists of the input and the states of the ROTPEN nonlinear plant model representing as $u(k)$ and $\boldsymbol{x}(k), \boldsymbol{x}(k+1)$ respectively. Assuming that the feedback linearized input vector is considered as $\boldsymbol{x}^{k+1} \coloneqq \boldsymbol{v}^{k}$. The control input of the nonlinear system might be written as $u^{k} := \Phi(\boldsymbol{x}^{k}, \boldsymbol{v}^{k})$ borrowed from Equation 3.5 where a nonlinear function $\Phi(\circ): \boldsymbol{R}^{nxm} \rightarrow \boldsymbol{R}$ with states and feedback linearized inputs of the nonlinear system.

As for implementation of LFL based algorithm with ANN, the LFL block might be formed with a suitable MLP-ANN possessing 2 hidden layers (Figure 4.4). Training set data of the LFL block is obtained by using $u(k)$ and $\Theta(k)$ depicted in Figure 4.5 with 0.001s sampling time. For training, inputs-output of the MLP are formed $[\boldsymbol{x}_{train}\ \boldsymbol{v}_{train}]^{T}$ and u_{train}, respectively (Figure 4.6). For testing the accuracy of the training of MLP, "*goodnessOfFit"* function is computed as 1 and it is used for test and test prediction data of the MLP output in terms of normalized MSE.

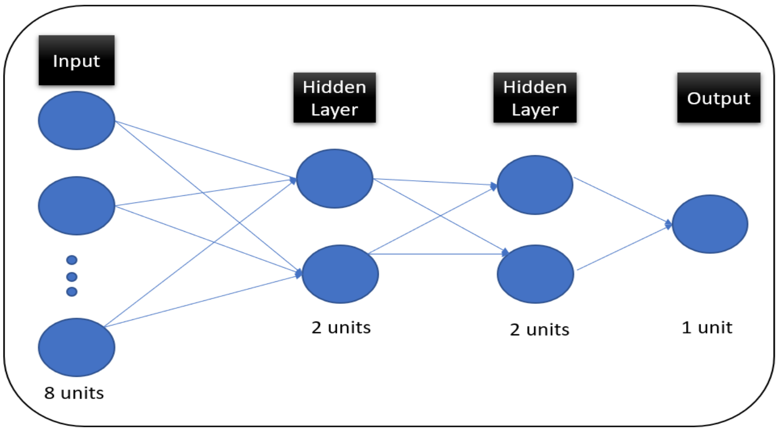
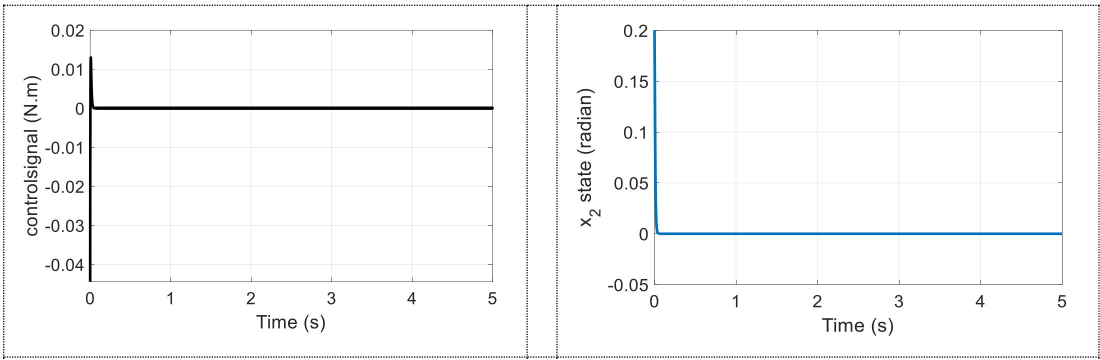


Figure 4.4. MLP feedforward network structure with "*nntraintool*".



(a) (b)

Figure 4.5. Training data set example $u(k)$ and $\theta(k)$.

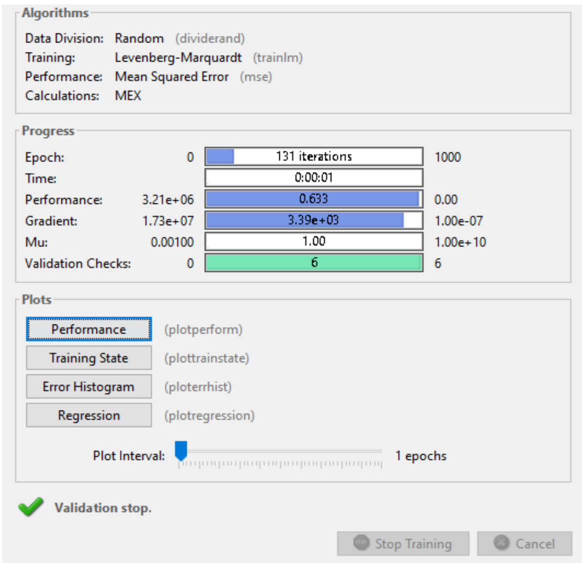


Figure 4.6. Training performance results.

As a result of LFL block training stage, it has got a nonlinear transformation providing an approximate feedback linearized system from input $\boldsymbol{v}(k)$ to the state $\boldsymbol{x}(k)$ in Fig. 4.7.

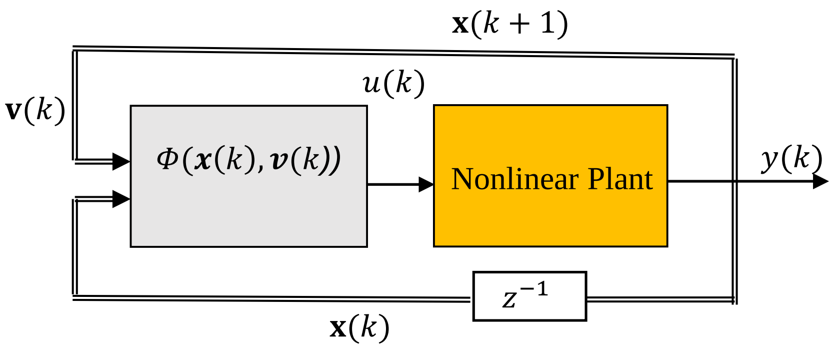


Figure 4.7. Feedback Linearized ROTPEN model with LFL.

4.2.2 Plant identification of the LFL based ROTPEN model via ARMA model

Choosing initial values of ARMA model parameters is a considerably complex issue for the online mode of the stable adaptive controller algorithm in terms of data-depended controller design algorithms. Therefore, initial values of the ARMA plant model parameters standing for a_n and b_n are firstly computed in a batch mode where the input-output data pairs chosen as $v^{lc}(k)$ and $y(k)$ which can be seen in Figure 4.8. The plant identification ARMA model parameters are determined by minimizing the identification error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.2 with MATLAB optimization toolbox function "fmincon" in terms of the time interval of $[k, k-K+1]$ for the LFL based ARMA plant model (Figure 4.8) which was also given in detail in Chapter 3.

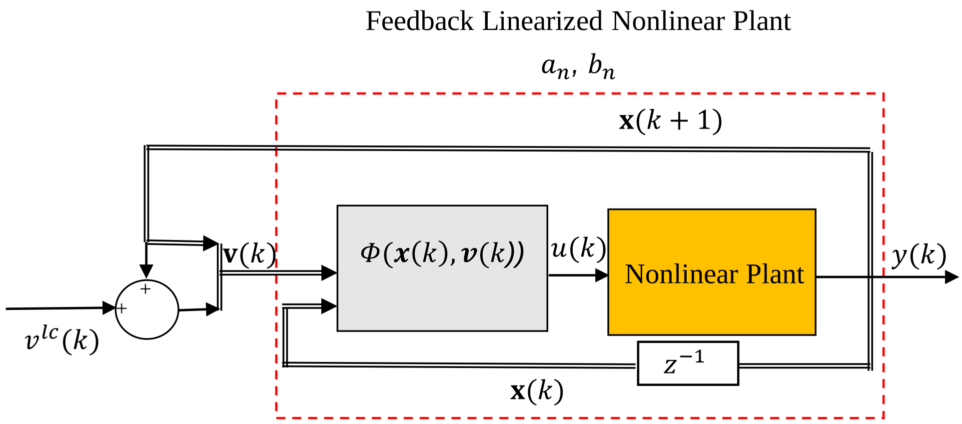


Figure 4.8. The developed LFL based ARMA plant model parameters for a_n and b_n.

$$\frac{1}{K}\sum_{s=0}^{K-1} \ell_{1,\varepsilon}\left(y_a(k-s), \sum_{n=1}^{N} a_n y(k-s-n) + \sum_{n=0}^{N} b_n v^{lc}(k-s-n)\right) + \lambda \left\| \begin{matrix} a \\ b \end{matrix} \right\|_2^2 \quad (4.2)$$

where N which stands for the degree of the ARMA plant parameters are to be determined in terms of the goodness of fit. Hence, ARX models of system identification toolbox of MATLAB environments are used for choosing appropriate N value with the simulated plant model and real plant data separately in terms of $v^{lc}(k)$ and $y(k)$. The performances of the ARX model fitness are tested with Akaike's information criterion (AIC) values computed in Table 4.2. AIC provides a measure of model quality obtained by simulating the situation where the candidate models are tested on different ARX degree. According to Akaike's theory, the most accurate model might be selected with the smallest AIC value [85].

Table 4.2. Comparison model fitness and AIC results of ARX models.

	For simulation y, v^{lc}		For real system y, v^{lc}	
ARX(nnk)* model	Model fitness (%)	AIC	Model fitness	AIC
arx000	-0.006636	-6.7234	-4.599	-6.3222
arx110	68.36	-12.7926	16.02	-11.4805
arx220	**100**	**-78.2319**	29.89	-11.5667
arx330	100	-70.4859	33.65	-12.9639
arx440	100	-74.2791	**34.61**	**-12.9853**
arx550	100	-75.4244	**34.51**	**-13.0960**

nnk*: Herein, first n denotes number of poles, the second n denotes number of zeros and k— Number of input samples that occur before the input affects the output, also called the *dead time* in the system.

In the light of Table 4.2 results, the fitness value of the arx220 model exactly matches the desired fitness value with the lowest AIC value for simulation environment whereas the fitness value of the arx550 model is acceptable value (when compared the other arx440 fitness result) with the nearly same AIC value for real plant. It is obvious that the simulation platform does not actually reflect the real system behaviors. However, according to evaluation of these results, the model degree of the ARMA plant model might be accepted as $N = 5$ for considering the real system.

As for online mode, determination of initial values of plant parameters is deduced with batch mode results for plant parameters a_n and b_n given in Table 4.3 where $\varepsilon = 0$ (ε-insensitive effects are especially analyzed for the real ROTPEN system response given the following sub-chapter), $K = 25, \lambda = 0.1$ which are chosen. Time evolutions of the developed LFL based NARMA controller for plant parameters both a_n and b_n are depicted in Figure 4.9.

Table 4.3. Initial parameters values of both a_n and b_n for the online mode obtained from the batch mode.

n	a_n	b_n
1	$8.28999730491958\times 10^{-5}$	-0.00637237736981552
2	$6.26130046926439\times 10^{-5}$	-0.00587480021640299
3	$4.12367167419285\times 10^{-5}$	-0.00533346235614781
4	$1.88134342900327\times 10^{-5}$	-0.00474868815420780
5	$-4.60320427712306\times 10^{-6}$	-0.00412098420370303
6		-0.00345104996739003

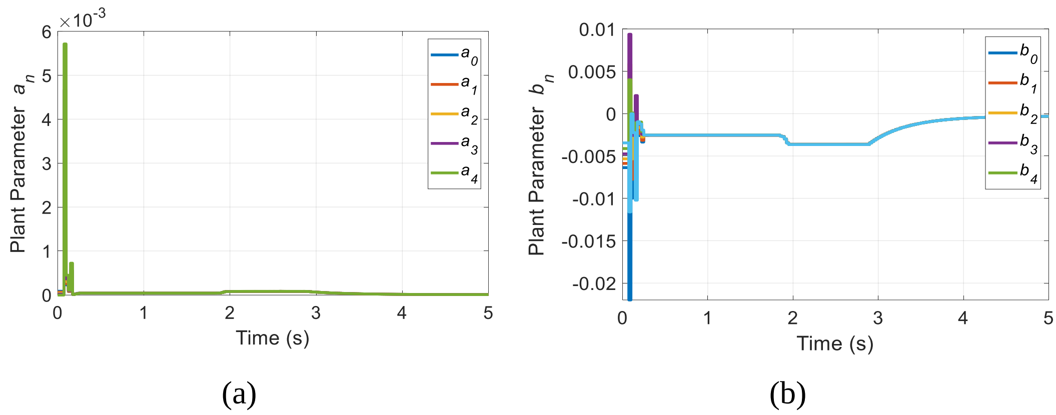


Figure 4.9: Time evolutions of the developed LFL based ARMA plant parameters for a_n and b_n in online mode.

4.2.3 Designing of the stable adaptive closed-loop system

To design the stable adaptive closed-loop system (Figure 4.10) in terms of finding appropriate the proposed controller in online mode, all the initial plant parameters and the closed-loop parameters of the ARMA models are determined in terms of a_n, b_n, α_n, and β_n providing c_m, and d_m within $L = 477$, $\lambda = 0.075$, and $N = 5$ in a batch mode within 500 samples. The controller algorithm is minimized with the tracking error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.4 with MATLAB optimization toolbox function "*fmincon*" in terms of the time interval of $[k, k - L + 1]$ with sliding window as L. These equations are also given with mathematical derivations in detail in Chapter 3. The proposed adaptive controller parameters are calculated as c_m, d_m and f_m obtained from a_n, b_n and α_n, β_n computed parameters using Equation 4.2 and 4.3. The obtained results of the parameters c_m, d_m and α_n, β_n are given in Table 4.4 and 4.5.

$$\frac{1}{L}\sum_{s=0}^{L-1} \ell_1(y_d(k-s), \sum_{n=0}^{2N} \alpha_n y(k-s-n) + \sum_{n=0}^{2N} \beta_n r(k-s-n)) + \lambda \left\|\begin{matrix}\alpha \\ \beta\end{matrix}\right\|_2^2 \quad (4.3)$$

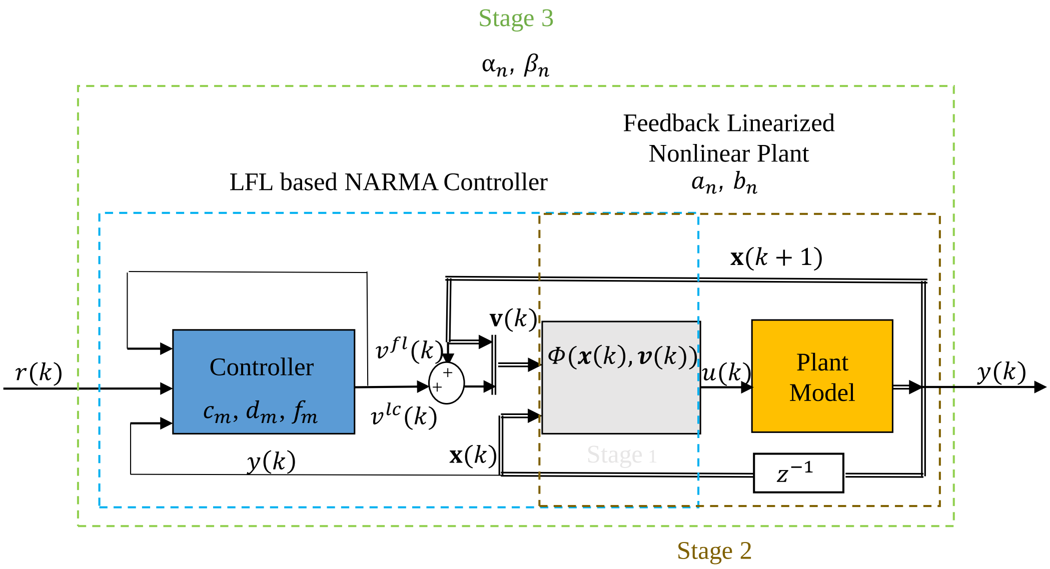


Figure 4.10: The proposed LFL based stable adaptive controller.

Table 4.4. Initial parameters values of the closed loop system as α_n and β_n obtained from batch mode.

n	α_n	β_n
0	0.00268950092851661	-8.96283217244484× 10^{-9}
1	0.00203025231244229	8.96340233196545× 10^{-9}
2	0.00165474545606356	8.96282882966701× 10^{-9}
3	0.00137678890586581	-8.96283221210843× 10^{-9}
4	0.00114826982244472	8.96282883471866× 10^{-9}
5	0.000949299138505982	-2.50858268700706× 10^{-8}
6	0.000769566600862029	-8.96283018141788× 10^{-9}
7	0.000602916971868847	-8.96283214721067× 10^{-9}
8	0.000445277110813857	-8.96283016434725× 10^{-9}
9	0.000293711388901573	-8.96283021128627× 10^{-9}
10	0.000145929991107909	8.96340237060842× 10^{-9}

Table 4.5. Initial parameters values of the proposed adaptive controller as c_m, d_m and f_m obtained from batch mode.

m	c_m	d_m	f_m
0	4.29109670285765× 10^{-6}	-0.470763708158013	-1
1	3.35063399748403× 10^{-7}	0.434036491043196	
2	3.90694941756246× 10^{-7}	-0.00203614442239089	
3	4.52751673809511× 10^{-7}	-0.00227096400042159	
4	5.21979941371609× 10^{-7}	-0.00253315835465237	
5	5.99204649850424× 10^{-7}	-0.00282565856410013	

During the online mode, the sliding window length, the degree and the regularization parameter of the proposed adaptive controller are chosen as $L = 50$, $N = 5$, and $\lambda = 0.1$, respectively. The tracking error performance of the closed-loop system is obtained by minimizing the tracking error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3 in terms of the time interval of $[k, k - L + 1]$. The ARMA controller parameters as c_m, and d_m are calculated by using Equation 4.3, 4.4 and 4.5 according to Diophantine equations [34].

$$v^{lc}(k) = \sum_{m=1}^{P} f_m v^{lc}(k-m) + \sum_{m=0}^{R} c_m r(k-m) + \sum_{m=0}^{Q} d_m y(k-m) \qquad (4.4)$$

$$a_o \coloneqq 1 + a_0\, f_0 \text{-} b_0\, d_0 \qquad (4.5)$$

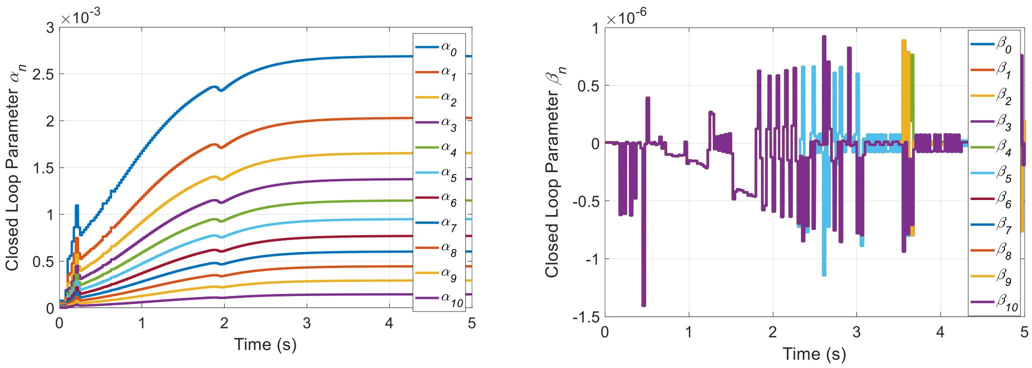
$$a_i \coloneqq \sum_{j=0}^{i} a_j f_{i-j} - \sum_{j=0}^{i} b_j d_{i-j} \text{ for } i \in \{1,2,\dots,N\}$$

$$a_i \coloneqq \sum_{j=i-N}^{N} a_j f_{i-j} - \sum_{j=i-N}^{N} b_j d_{i-j} \; for \; i \in \{N+1, N+2, \dots, 2N\}$$

$$\beta_i \coloneqq -\sum_{j=0}^{i} b_j c_{i-j} \text{ for } i \in \{0,1,2,\dots,N\}$$

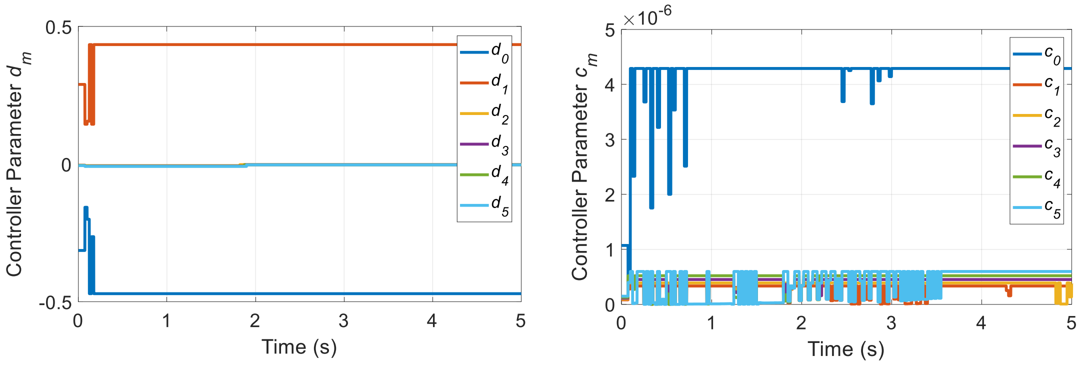
$$\beta_i \coloneqq -\sum_{j=i-N}^{N} b_j c_{i-j} \text{ for } i \in \{N+1, N+2, \dots, 2N\}$$

Time evolutions of the developed LFL based adaptive controller for closed-loop parameters as α_n, β_n, the adaptive controller parameters as c_m, d_m, the controller signal as v_2^{lc}, and the rod angle of the ROTPEN as $y \coloneqq \theta$ are depicted as in Figure 4.11, 4.12 and 4.13, respectively.



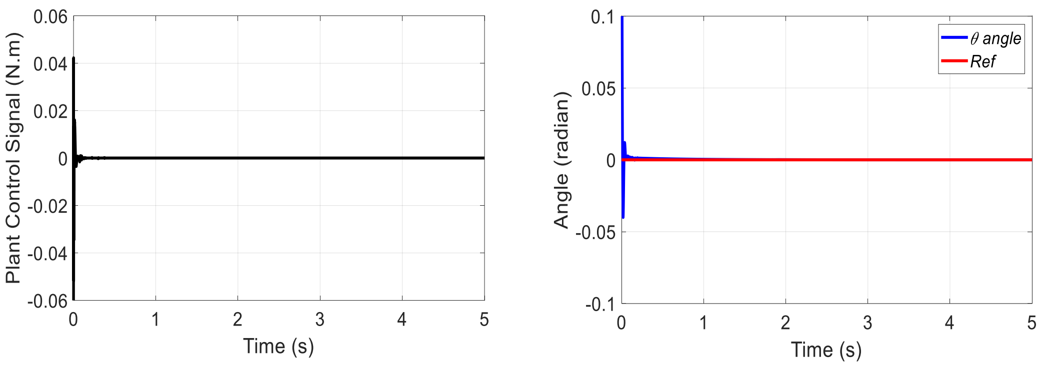
(a) (b)

Figure 4.11: Time evolutions of the developed closed-loop system parameters a) α_n and b) β_n.



(a) (b)

Figure 4.12: Time evolutions of the developed LFL based adaptive controller parameters a) d_m and b) c_m.



(a) (b)

Figure 4.13: Time evolutions of the controller signal a) v_2^{lc} and the rod angle b) $\coloneqq \theta$.

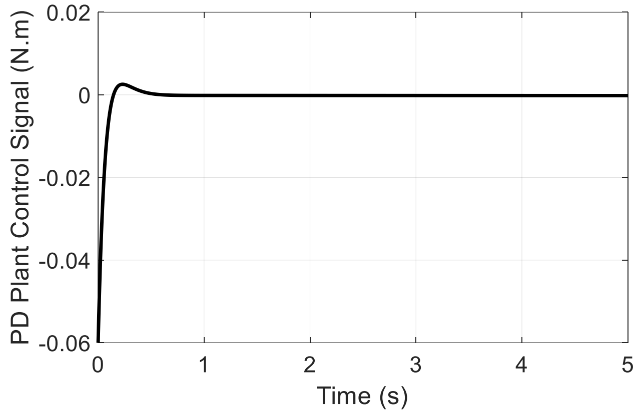
4.2.4 Simulation results comparisons of the PD controller and the proposed controller performances for ROTPEN

The performances of both the proposed adaptive controller and the PD controller are compared each other. The PD controller design is represented in Equation 4.6 with the reference book of the experimental setup where the controller parameters K_p and K_d are taken as 80 and 10.5, respectively from [44]. The analysis for designing a digital implementation of a PD controller in MATLAB/SIMULINK is implemented by according to the standard form of the PID controller to be discretized. Approximations for first-order derivatives are made by backward finite differences. The integral term is discretized, with a sampling time Δt, as follows: $\int_0^{t_k} e(\tau)d\tau = \sum_{i-1}^{k} e(t_i)\Delta t$. The derivative term is approximated as, $\frac{de(t_k)}{dt} = \frac{e(t_k) - e(t_{k-1})}{\Delta t}$. Thus, a velocity algorithm for implementation of the discretized PID controller obtained by differentiating $u(t)$ using the numerical definitions of the first and second derivative and solving for $u(t_k)$ and finally obtaining:

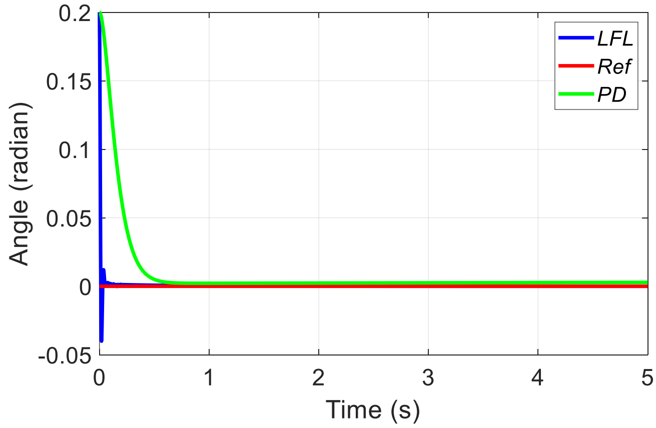
$$u(t_k) = u(t_{k-1}) + K_p\left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right)e(t_k) + \left(-1 - \frac{2T_d}{\Delta t}\right) e(t_{k-1}) + \frac{T_d}{\Delta t}e(t_{k-2})\right] \tag{4.6}$$

where $T_i = \frac{K_p}{K_i}$ and $T_i = \frac{K_d}{K_p}$ [87]. The controller performances are depicted in Figure 4.14 in terms of pendulum angle MSE. Performance evaluation criteria is used as MSE in Equation 4.7 where $e(k)$ stands for the closed-loop system tracking error and S stands for the number of samples. LFL based adaptive controller provides lower error according the MSE error in Table 4.6. As for the settling time evaluation, the settling time of the proposed controller is observed as 0.0467 seconds and it is less than PD controller's settling time. The minimum overshoot percentage of the proposed controller is a good response value as 4% which significantly less than 25% level which might be acceptable value [86] in Table 4.7.

$$MSE = \frac{1}{S}\sum_{k=1}^{S} e^2(k) \tag{4.7}$$



a)



b)

Figure 4.14: a) PD Plant Control Signal b) Performance comparison of the proposed LFL based adaptive controller and the PD controller.

Table 4.6. Performance evaluation of PD and the proposed LFL based controller in terms of MSE.

Controller	MSE
PD	8.59×10^{-4}
LFL Based Adaptive	4.29×10^{-5}

Table 4.7. Performance evaluation of PD and the proposed LFL based controller.

Controller	Settling Time (s)	Percentage Overshoot
PD	0.4582	0
LFL Based Adaptive	0.0467	4

4.2.5 The ε-insensitive and λ-regularization parameters analysis of loss function

The loss function of the closed-loop system identification is considered as ε −insensitive $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3. To test the robustness performance of the proposed LFL based robust adaptive controller, a white noise which has a 2 dB 'SNR' is added to the plant control signal of the closed-loop system via MATLAB "AWGN" function (Figure 4.15). The minimization of tracking error is tested with different ε values and the MSE results of the loss function are given in Table 4.8 values with and w/o noise. Moreover, the different λ regularization parameters of the closed loop system tracking error called as loss function are tested for $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3. The obtained tracking error MSE results of the proposed LFL based adaptive controller are represented in Table 4.9.

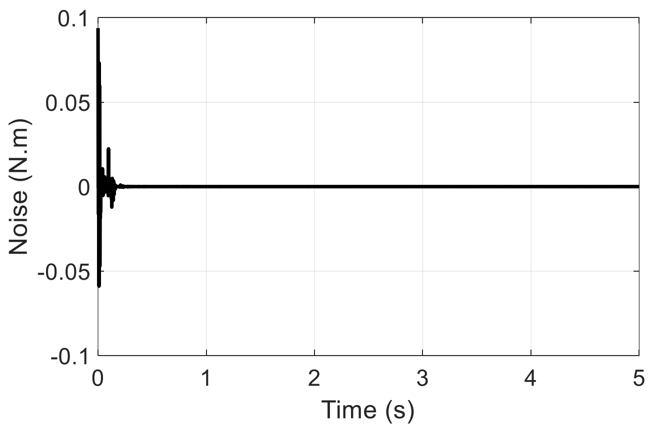


Figure 4.15: The generated noise signal with AWGN function.

Table 4.8. Performance evaluation of ε −insensitive with and w/o noise in Equation 4.3 for the proposed LFL based adaptive controller.

ε	MSE (With Noise)	MSE (Without Noise)
0	7.7448×10^{-5}	4.2907×10^{-5}
0.0001	$\mathbf{7.7395 \times 10^{-5}}$	4.2930×10^{-5}
0.001	7.7519×10^{-5}	$\mathbf{4.2880 \times 10^{-5}}$
0.01	7.5844×10^{-5}	4.2964×10^{-5}
0.1	7.5844×10^{-5}	4.2964×10^{-5}

Table 4.9. Performance evaluation of λ regularization in Equation 4.3 for the proposed LFL based adaptive controller.

λ	MSE
0.05	4.2954×10^{-5}
0.075	4.2912×10^{-5}
0.1	$\mathbf{4.29 \times 10^{-5}}$
0.35	4.2937×10^{-5}
0.6	4.2930×10^{-5}

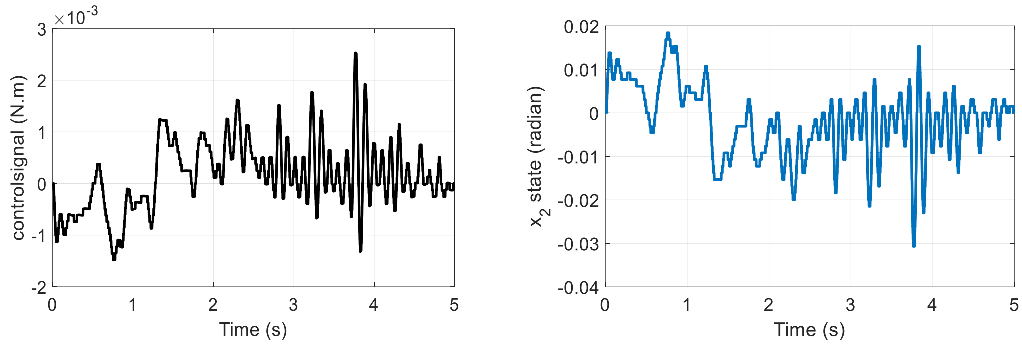
4.3. The Experimental Results of the Proposed Controller for ROTPEN

The proposed adaptive controller is tested on physical ROTPEN plant via SIMULINK environment. The proposed algorithm is achieved by three progressive stages as follows; i) NARMA based LFL strategy is used to obtain a feedback linearized nonlinear plant by using ANN, ii) the NARMA-LFL based feedback linearized plant might be identified as an ARMA plant model with ε-insensitive loss function for system identification, and iii) the overall closed-loop control system providing Schur stability conditions and ε-insensitive loss function for tracking error is constituted by both ARMA plant and controller model. These stages of the experimental studies are explained in the following subsections.

4.3.1 LFL for real ROTPEN system

To achieve NARMA based LFL via MLP for the ROTPEN plant, training data set is formed. The data set consists of the input and the states of the ROTPEN nonlinear plant model representing as $u(k)$ and $\boldsymbol{x}(k)$, $\boldsymbol{x}(k+1)$ respectively. Assuming that the feedback linearized input vector is considered as $\boldsymbol{x}^{k+1} \coloneqq \boldsymbol{v}^{k}$. The control input of the nonlinear system might be written as $u^{k}:= \Phi(\boldsymbol{x}^{k}, \boldsymbol{v}^{k}):= \Phi(\boldsymbol{x}^{k}, \boldsymbol{x}^{k+1})$ borrowed from Equation 3.5 where a nonlinear function $\Phi(\circ): \boldsymbol{R}^{nxm} \rightarrow \boldsymbol{R}$ with states and feedback linearized inputs of the nonlinear system.

As for implementation of LFL based algorithm with ANN, the LFL block might be formed with a suitable MLP-ANN possessing 2 hidden layers (Figure 4.4). Training set data of the LFL block is obtained by using $u(k)$ and $\theta(k)$ depicted in Figure 4.16 with 0.001s sampling time. For training, inputs-output of the MLP are formed $[\boldsymbol{x}_{train}\ \boldsymbol{v}_{train}]^T$ and u_{train}, respectively (Figure 4.17). For testing the accuracy of the training of MLP, "*goodnessOfFit*" function is computed as 1 and it is used for test and test prediction data of the MLP output in terms of normalized MSE. As a result of LFL block training stage, it has got a nonlinear transformation providing an approximate feedback linearized system from input $\boldsymbol{v}(k)$ to the state $\boldsymbol{x}(k)$ in Fig. 4.7.



(a) (b)

Figure 4.16: Training data set example $u(k)$ and $\theta(k)$.

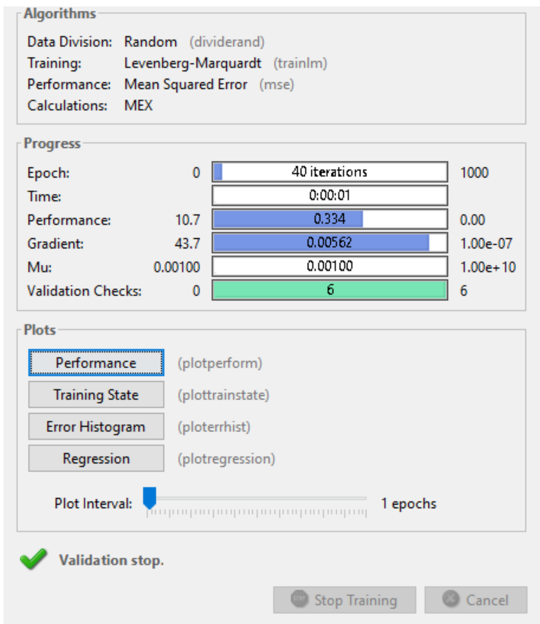


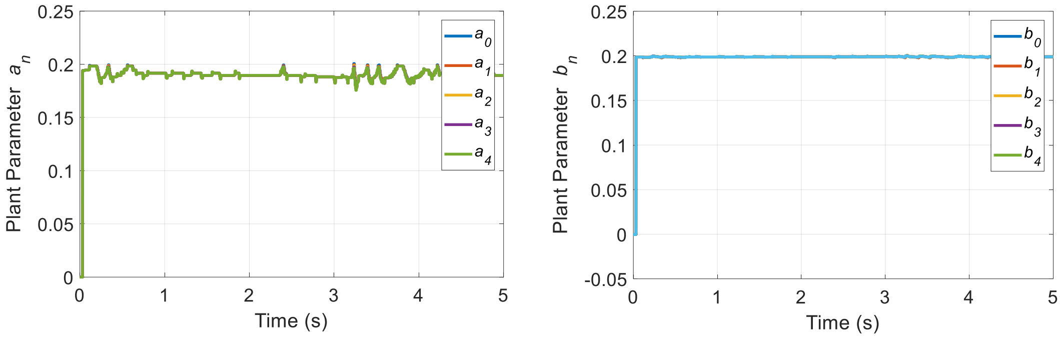
Figure 4.17: Training performance results.

4.3.2 Real plant identification of the LFL based ROTPEN system via ARMA model

Initial values of the ARMA plant model parameters standing for a_n and b_n are firstly computed in a batch mode where the input-output data pairs chosen as $v^{lc}(k)$ and $y(k)$ which can be seen in Figure 4.7. The plant identification ARMA model parameters are determined by minimizing the identification error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.2 with "user defined gradient optimization" via SIMULINK in terms of the time interval of $[k, k-K+1]$ for the LFL based ARMA plant model. As for online mode, determination of initial values of plant parameters is deduced with batch mode results for plant parameters a_n and b_n given in Table 4.10 where $K = 377$, $\lambda = 0.075$ and $N = 5$ which are chosen. Time evolutions of the developed LFL based NARMA controller for plant parameters both a_n and b_n are depicted in Figure 4.18.

Table 4.10. Initial parameters values of both a_n and b_n for the online mode obtained from the batch mode.

n	a_n	b_n
1	$6.80282031888859\times 10^{-6}$	-0.000315314950195759
2	$6.80439378482749\times 10^{-6}$	-0.000315386533787749
3	$6.80597087464735\times 10^{-6}$	-0.000315458278908380
4	$6.80755152705512\times 10^{-6}$	-0.000315530185998794
5	$6.80913564732295\times 10^{-6}$	-0.000315602255297558
6		-0.000315674487315689



(a) (b)

Figure 4.18. Time evolutions of the developed LFL based ARMA plant parameters for a_n and b_n in online mode.

4.3.3 Designing of the real plant based stable adaptive closed-loop system

To design the stable adaptive closed-loop system (Figure 4.9) in terms of finding appropriate the proposed controller in online mode, all the initial plant parameters and the closed-loop parameters of the ARMA models are determined in terms of a_n, b_n, α_n, and β_n providing c_m, and d_m within $L = 477$, $\lambda = 0.075$, and $N = 5$ in a batch mode. The controller algorithm is minimized with the tracking error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3 with "user defined gradient optimization" via SIMULINK in terms of the time interval of $[k, k-L+1]$ with sliding window as L. Herein, MATLAB optimization toolbox function "fmincon" could not be used because this function cannot be built by the MATLAB environment so the "user defined subgradient optimization" algorithm is coded with C program. The proposed adaptive controller parameters are calculated as c_m, d_m and f_m obtained from a_n, b_n and α_n, β_n computed parameters using Equation 4.4. The obtained results of the parameters c_m, d_m and α_n, β_n are given in Table 4.11 and 4.12.

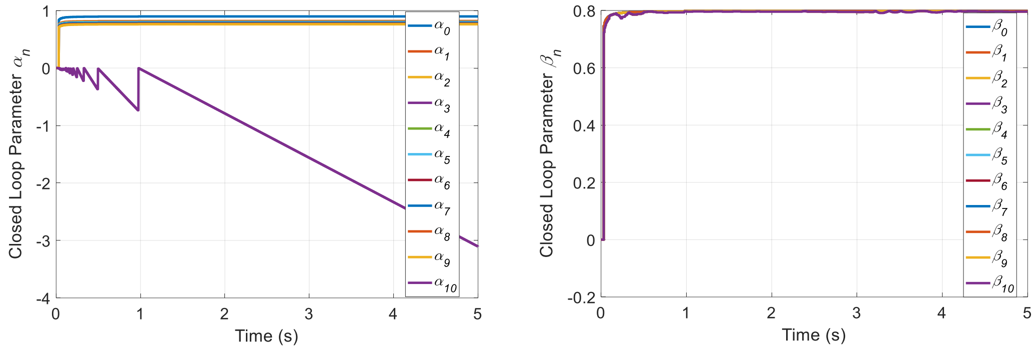
Table 4.11. Initial parameters values of the closed loop system as α_n and β_n obtained from batch mode.

n	α_n	β_n
0	0.00268950092851661	$-8.96283217244484\times 10^{-9}$
1	0.00203025231244229	$8.96340233196545\times 10^{-9}$
2	0.00165474545606356	$8.96282882966701\times 10^{-9}$
3	0.00137678890586581	$-8.96283221210843\times 10^{-9}$
4	0.00114826982244472	$8.96282883471866\times 10^{-9}$
5	0.000949299138505982	$-2.50858268700706\times 10^{-8}$
6	0.000769566600862029	$-8.96283018141788\times 10^{-9}$
7	0.000602916971868847	$-8.96283214721067\times 10^{-9}$
8	0.000445277110813857	$-8.96283016434725\times 10^{-9}$
9	0.000293711388901573	$-8.96283021128627\times 10^{-9}$
10	0.000145929991107909	$8.96340237060842\times 10^{-9}$

Table 4.12. Initial parameters values of the proposed adaptive controller as c_m, d_m and f_m obtained from batch mode.

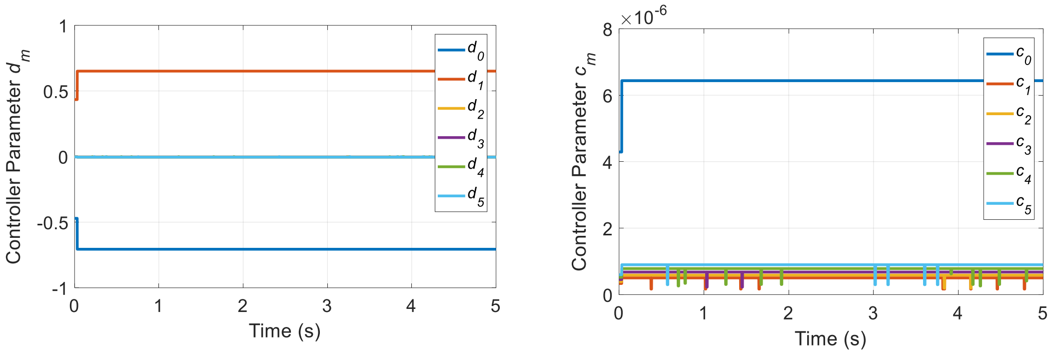
m	c_m	d_m	f_m
0	$4.29109670285765\times 10^{-6}$	-0.470763708158013	-1
1	$3.35063399748403\times 10^{-7}$	0.434036491043196	
2	$3.90694941756246\times 10^{-7}$	-0.00203614442239089	
3	$4.52751673809511\times 10^{-7}$	-0.00227096400042159	
4	$5.21979941371609\times 10^{-7}$	-0.00253315835465237	
5	$5.99204649850424\times 10^{-7}$	-0.00282565856410013	

During the online mode, the sliding window length, the degree and the regularization parameter of the proposed adaptive controller are chosen as $L = 20$, $N = 5$, and $\lambda = 0.075$, respectively. The tracking error performance of the closed-loop system is obtained by minimizing the tracking error defined with ε-insensitive loss function $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3 in terms of the time interval of $[k, k-L+1]$. The ARMA controller parameters as c_m, and d_m are calculated by using Equation 4.3, 4.4 and 4.5 according to Diophantine equations [34]. Time evolutions of the developed LFL based adaptive controller for closed-loop parameters as α_n, β_n, the adaptive controller parameters as c_m, d_m, the controller signal, and the rod angle of the ROTPEN as $y \coloneqq \theta$ are depicted as in Figure 4.19, 4.20 and 4.21, respectively.



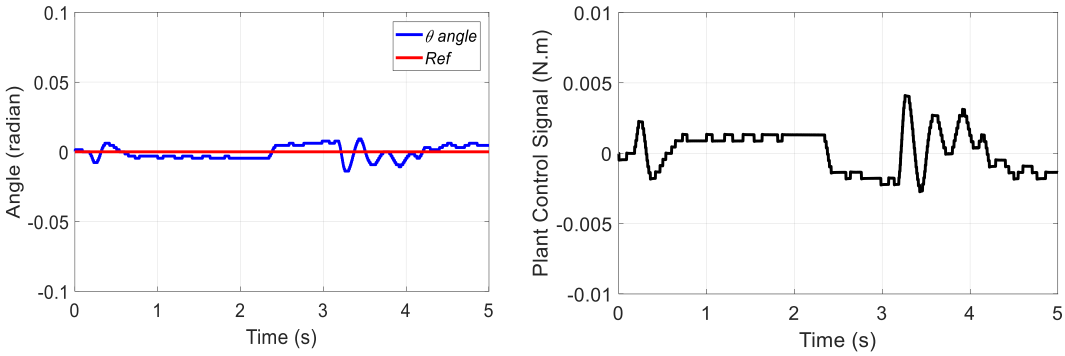
(a) (b)

Figure 4.19. Time evolutions of the developed closed-loop system parameters a) α_n and b) β_n.



(a) (b)

Figure 4.20: Time evolutions of the developed LFL based adaptive controller parameters a) d_m and b) c_m.

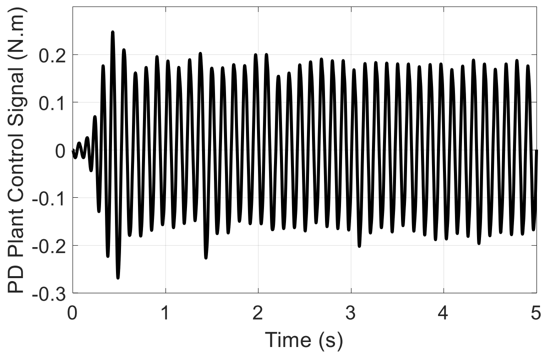


(a) (b)

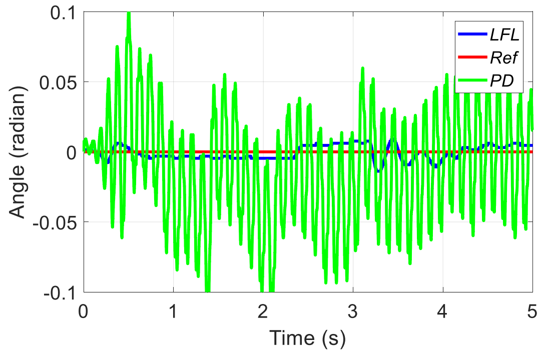
Figure 4.21: The proposed controller's a) performance and b) signal

4.3.4 Experimental results comparison of the PD controller and the proposed controller for ROTPEN

The performances of both the proposed adaptive controller and the PD controller are compared for real ROTPEN system each other. The PD controller design is represented in Equation 4.6 with the reference book of the experimental setup where the controller parameters K_p and K_d are taken as 80 and 10.5, respectively from [44]. The controller performances and PD controller signal are depicted in Figure 4.22. LFL based adaptive controller provides lower error according the MSE error in Table 4.13. As for the settling time evaluation, the settling time of the proposed controller is observed as 0.091 seconds. The minimum overshoot percentage of the proposed controller is a good response value as 0.15% which significantly less than 25% level which might be acceptable value [86] in Table 4.14.



(a)



(b)

Figure 4.22: a) PD control signal b) comparison of PD and the proposed controller.

Table 4.13. Performance evaluation of PD and the proposed LFL based controller in terms of MSE.

Controller	MSE
PD	0.0018
LFL Based	2.4801×10^{-5}

Table 4.14. Performance evaluation of PD and the proposed LFL based controller for real ROTPEN system.

Controller	Settling Time	Percentage Overshoot
PD	0.05	0.7
LFL Based NARMA	0.091	0.15

The controller parameters K_p and K_d are taken as 80 and 2. PD plant control signal and the controller performances are depicted in Figure 4.23, Figure 4.24, respectively. LFL based adaptive controller provides lower error according the MSE error in Table 4.15. As for the settling time evaluation, the settling time of the proposed controller is observed as 0.091 seconds and it is less than PD controller's settling time. The minimum overshoot percentage of the proposed controller is a good response value as 0.15% which significantly less than 25% level which might be acceptable value [86] in Table 4.16.

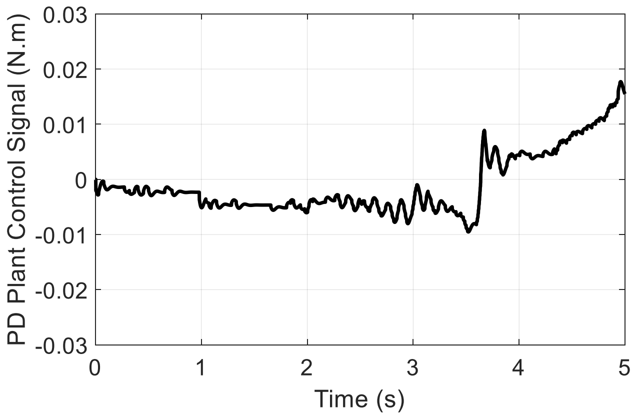


Figure 4.23. PD control signal

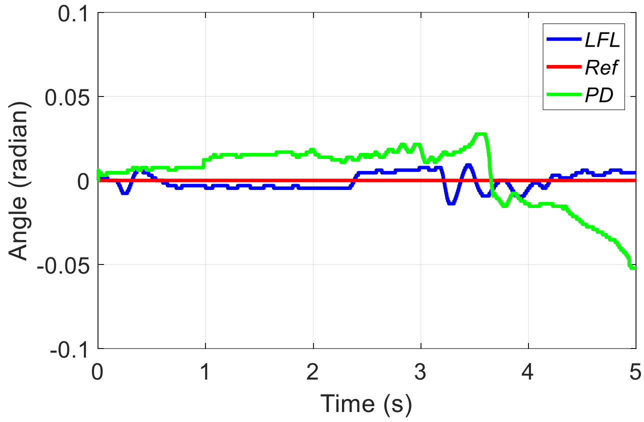


Figure 4.24. Comparison of PD and the proposed controller.

Table 4.15. Performance evaluation of PD and the proposed LFL based controller.

Controller	MSE
PD	7.0587×10^{-5}
LFL Based	2.4801×10^{-5}

Table 4.16. Performance evaluation of PD and the proposed LFL based controller for real ROTPEN system.

Controller	Settling Time	Percentage Overshoot
PD	0.5	1.3
LFL Based NARMA	0.091	0.15

4.3.5 The ε-insensitive and λ-regularization parameters analysis of loss function

The loss function of the closed-loop system identification is considered as ε −insensitive $\ell_{1,\varepsilon}(\cdot,\cdot)$ given in Equation 4.3. To test the robustness performance of the proposed LFL based robust adaptive controller, a white noise which has a 2 dB ‘SNR’ is added to the plant control signal of the closed-loop system via SIMULINK’s “AWGN” block (Figure 4.25). The minimization of tracking error is tested with different ε values and the MSE results of the loss function are given in Table 4.17 values with and w/o noise. Moreover, the different λ regularization parameters of the closed loop system tracking error called as loss function are tested for $\ell_{1,\varepsilon}$ given in Equation 4.3. The obtained tracking error MSE results of the proposed LFL based adaptive controller are represented in Table 4.18.

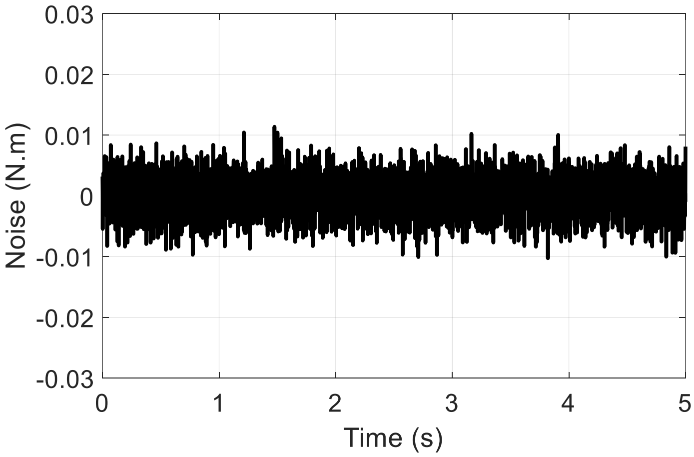


Figure 4.25. The generated noise signal with AWGN block.

Table 4.17. Performance evaluation of ε −insensitive with and w/o noise in Equation 4.3 for the proposed LFL based adaptive controller applying to the real ROTPEN system.

ε	MSE (With Noise)	MSE (Without Noise)
0	9.5764×10^{-4}	2.4801×10^{-5}
0.0001	3.7541×10^{-4}	9.7199×10^{-5}
0.001	$3.6333e \times 10^{-4}$	$\mathbf{2.1595 \times 10^{-5}}$
0.01	$\mathbf{2.7482 \times 10^{-5}}$	8.4007×10^{-5}
0.1	2.9778×10^{-5}	5.9850×10^{-4}

Table 4.18. Performance evaluation of λ regularization parameter in Equation 4.3 for the proposed LFL based adaptive controller applying to the real ROTPEN system.

λ	MSE
0.025	1.1791×10^{-4}
0.05	5.5302×10^{-5}
0.075	$\mathbf{2.4801 \times 10^{-5}}$
0.1	6.3048×10^{-5}

5. CONCLUSIONS

In this thesis, the proposed algorithm which is LFL based stable adaptive NARMA controller is achieved for the ROTPEN. The proposed stable robust adaptive control algorithm is implemented via the ARMA models of both the plant and the controller The controller's design scheme possesses three design stages ; i) NARMA based LFL is used to obtain a feedback linearized model for a nonlinear plant by using the artificial neural network (ANN), ii) the NARMA-LFL based plant might be identified as an auto-regressive moving average (ARMA) plant model, and iii) the closed-loop control system providing Schur stability conditions is constituted by both ARMA plant and controller models. After controller design, the overall closed-loop system is obtained as a linear dynamical system with possessing Schur stability. To provide robustness, ε-insensitive loss functions in the identification and controller design phases are used.

The proposed adaptive controller design scheme is tested on a simulated ROTPEN for angular rod position and compared to PD controller in terms of MSE for tracking performance, the overshoot and settling time. MSE values of PD and LFL based adaptive controller are respectively computed as 8.59×10^{-4} and 4.29×10^{-5} in terms of tracking error of the closed loop system. LFL based adaptive controller settling time is observed as 0.0467s which is nearly 10% of the PD one. Furthermore, different ε −values are evaluated against with and without noise for $\ell_{1,\varepsilon}(\cdot,\cdot)$ loss function of tracking error updating the adaptive controller parameters. According to the minimum MSE values, ε −values are determined as 0.0001 and 0.001 for with noise and without noise, respectively.

When the proposed controller is tested on a real ROTPEN system for angular rod position and compared to PD controller in terms of MSE for tracking performance, the overshoot and settling time. MSE values of PD and LFL based adaptive controller are respectively computed as 0.0018 and 2.4801×10^{-5} in terms of tracking error of the closed loop system. LFL based adaptive controller percentage overshoot is observed as 0.15 which is nearly 20% of the PD one. Moreover, different ε −values are evaluated against with and without noise for $\ell_{1,\varepsilon}(\cdot,\cdot)$ loss function of tracking error updating the adaptive controller parameters. According to the minimum MSE values, ε −values are determined as 0.01 and 0.001 for with noise and without noise, respectively. According to minimum MSE value, λ regularization parameter is determined as 0.075 for the proposed LFL based adaptive controller.

The comparison of simulation and real system results shows the potential of the proposed LFL based stable adaptive NARMA controller. For future works, SISO ARMA modelling can be developed to MIMO ARMA modelling for the proposed controller

REFERENCES

1. Grizzle JW, Kokotovic PV. Feedback linearization of sampled-data systems. IEEE Transactions on Automatic Control. 1988; 33(9): 857-859.
2. Khalil HK. Nonlinear Systems. 2nd ed.: Prentice Hall; 1996.
3. Sastry SS. Nonlinear Systems Analysis, Stability and Control. 1st ed.: Springer-Verlag New York; 1999.
4. Sastry SS, Isidori A. Adaptive control of linearizable systems. IEEE Transactions on Automatic Control. 1989; 34(11): 1123-1131.
5. Jean-Jacques E. Slotine , Li W. Applied Nonlinear Control; 1991.
6. Vidyasagar M. Nonlinear Systems Analysis. 2nd ed.: Prentice Hall; 1992.
7. Conte G, Moog CH, Perdon AM. Nonlinear Control Systems. 1st ed.: Springer-Verlag London; 1999.
8. Hunt LR, Su R, Meyer G. Approximating linearizations for nonlinear systems. Circuits, Systems and Signal Processing. 1985; 5(4): 419-433.
9. Isidori A. Nonlinear Control Systems. 3rd ed.: Springer; 1995.
10. Krener AJ. Feedback Linearization. In Krener AJ. Mathematical control theory. New York: Springer; 1999. p. 66-98.
11. Sontag ED. Mathematical Control Theory. 1st ed.: Springer-Verlag New York; 1990.
12. Spong MW. Partial feedback linearization of underactuated mechanical systems. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94); 1994; Munich: IEEE XPLORE.
13. III FJD. An anti-windup input–output linearization scheme for SISO systems. Journal of Process Control. 1999; 9(3): 213-220.
14. Fuh CC, Tsai HH. Control of discrete-time chaotic systems via feedback linearization. Chaos, Solitons & Fractals. 2002; 13(2): 285-294.
15. Marquez-Martinez LA, Moog CH. Input-output feedback linearization of time-delay systems. IEEE Transactions on Automatic Control. 2004; 49(5): 781-785.
16. Ho MT, Tu YW, Lin Hs. Controlling a ball and wheel system using full-state-feedback linearization [Focus on Education]. IEEE Control Systems. 2009; 29.
17. Zhou QL, Zhang Y, Rabbath CA, Theilliol D. Design of feedback linearization control and reconfigurable control allocation with application to a quadrotor UAV. In 2010 Conference on Control and Fault-Tolerant Systems (SysTol); 2010; Nice: IEEE.

18. Cambera JC, Feliu-Batlle V. Input-state feedback linearization control of a single-link flexible robot arm moving under gravity and joint friction. Robotics and Autonomous Systems. 2017; 88: 24-36.
19. Pathak K, Franch J, Agrawal SK. Velocity control of a wheeled inverted pendulum by partial feedback linearization. In 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601); 2004; Nassau: IEEE.
20. Pathak K, Franch J, Agrawal SK. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. IEEE Transactions on Robotics. 2005; 21(3): 505-513.
21. Yu H, Liu Y, Yang T. Closed-loop tracking control of a pendulum-driven cart-pole underactuated system. Proceedings of the Instution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 2008; 222(2): 109-125.
22. Türker T, Görgün H, Cansever G. Lyapunov's direct method for stabilization of the Furuta pendulum. Turkish Journal of Electrical Engineering and Computer Sciences. 2012; 20: 99-110.
23. Moreno-Valenzuela J, Aguilar-Avelar C. Feedback Linearization Control of Furuta Pendulum. In Moreno-Valenzuela J, Aguilar-Avelar C. Motion Control of Underactuated Mechanical Systems.: Springer International Publishing; 2018.
24. Lewis FL, Jagannathan S, Yeşildirek A. Neural network control of robot manipulators and nonlinear systems London: Taylor & Francis; 1999.
25. Şahin S. Learning Feedback Linearization Using Artificial Neural Networks. Neural Processing Letters. 2016; 44(3): 625-637.
26. Hagan MT, Demuth HB, Jesús OD. An introduction to the use of neural networks in control systems. International Journal of Robust and Nonlinear Control 12. 2002;: 959-985.
27. Dheeraj K, Jacob J, Nandakumar MP. Direct Adaptive Neural Control Design for a Class of Nonlinear Multi Input Multi Output Systems. IEEE Access. 2019; 7: 15424-15435.
28. He S, Unbehauen R. Approximate Feedback Linearisation Using Multilayer Neural Networks. Neural Processing Letters. 1998; 8(2): 131-144.
29. He S, Relf K, Unbehauen R. A neural approach for control of nonlinear systems with feedback linearization. IEEE Transactions on Neural Networks. ; 9(6): 1409-1421.
30. Qu D, Tang D. On control structure scheme of feedback linearization for nonlinear system based on ANN models and simulation researches. In 2010 3rd International Congress on Image and Signal Processing; 2010; Yantai.
31. Narendra KS, Mukhopadhyay S. Adaptive control using neural networks and approximate models. IEEE Transactions on Neural Networks. 1997; 8(3): 475-485.
32. Deng H, Li HX, Wu YH. Feedback-Linearization-Based Neural Adaptive Control for Unknown Nonaffine Nonlinear Discrete-Time Systems. IEEE Transactions on Neural Networks. 2008; 19(9): 1615-1625.
33. Pedro JO, Dahunsi OA. Neural network based feedback linearization control of a servo-hydraulic vehicle suspension system. Applied Mathematics and Computer Science. 2011; 21: 137-147.

34. Astrom KJ, Wittenmark B. Adaptive Control Boston: Addison-Wesley Longman Publishing Co., Inc. MA, USA; 1994.
35. Zhao Q, Xu H, Jagannathan S. Neural Network-Based Finite-Horizon Optimal Control of Uncertain Affine Nonlinear Discrete-Time Systems. IEEE Transactions on Neural Networks and Learning Systems. 2015 March; 26(3): 486-499.
36. Blanchini F, Parisini T, Pellegrino FA, Pin G. High-Gain Adaptive Control: A Derivative-Based Approach. IEEE Transactions on Automatic Control. 2009 September; 54(9): 2164-2169.
37. Battistelli G, Hespanha JP, Mosca E, Tesi P. Model-Free Adaptive Switching Control of Time-Varying Plants. IEEE Transactions on Automatic Control. 2013 May; 58(5): 1208-1220.
38. Waegeman T, Wyffels F, Schrauwen B. Feedback Control by Online Learning an Inverse Model. IEEE Transactions on Neural Networks and Learning Systems. 2012 October; 23(10): 1637-1648.
39. Wang FY, Jin N, Liu D, Wei Q. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound. IEEE Transactions on Neural Networks. 2011 January; 22(1): 24-36.
40. Zhu Y, Hou Z. Data-Driven MFAC for a Class of Discrete-Time Nonlinear Systems With RBFNN. IEEE Transactions on Neural Networks and Learning Systems. 2014 December; 25(5): 1013-1020.
41. Şahin S, Güzeliş C. Online Learning ARMA Controllers With Guaranteed Closed-Loop Stability. IEEE Transactions on Neural Networks and Learning Systems. 2016; 27(11): 2314-2326.
42. Tibshirani R. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996; 58(1): 267-288.
43. Kukreja SL, Löfberg J, Brenner MJ. A LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO) FOR NONLINEAR SYSTEM IDENTIFICATION. In IFAC Proceedings Volumes; 2006. p. 814-819.
44. QUANSER. QNET ROTPEN Workbook (Student).
45. Soydemir MU, Alkuş İ, Bulucu P, Kocaoğlu A, Güzeliş C, Şahin S. Data Dependent Stable Robust Adaptive Controller Design for Altitude Control of Quadrotor Model. In 2018 18th International Conference on Mechatronics-Mechatronika (ME); 2018; Brno.
46. Slotine, J.-J., & Li, W. (1991). *Applied Nonlinear Control.* Pearson
47. Maciejowski J. Predictive Control With Constraints: Pearson education; 2002.
48. Sjöberg J, Zhang Q, Ljung L, Benveniste A, Delyon B, Glorennec PY, et al. Nonlinear black-box modeling in system identification: a unified overview. Automatica (Journal of IFAC). 1995 December; 31(12): 1691-1724.
49. Cabrera JBD, Narendra KS. Issues in the application of neural networks for tracking based on inverse control. IEEE Trans. Automat. Contr. 1999; 44: 2007-2027.
50. Levin AU, Narendra KS. Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization. IEEE Transaction on Neural Networks. 1993; 4(2): 192-206.

51. Yegnanarayana B. Design of ARMA Digital Filters by Pole-Zero. IEEE Transactions on Acoustics, Speech and Signal Processing. 1981 June; 29(3): 433-439.
52. Brown RG. Smoothing, Forecasting and Prediction of Discrete Time: Prentice-Hall; 1994.
53. Chen S, A. BS. Modelling and Analysis of Non-linear Time Series. International Journal of Control. 1989; 50(6): 2151-2171.
54. Chen LJ, Narendra KS. Intelligent Contro1 Using Multiple Neural Networks. International Journal of Adaptive Control and Signal Processing. 2003; 17(6): 417-430.
55. Hyvärinen A, Oja E. A Fast Fixed-Point Algorithm for Independent Component Analysis. Neural Computation. 1997; 9(7): 1483-1492.
56. Rank E. Application of Bayesian Trained RBF Networks to Nonlinear Time-. Signal Processing. 2003; 83(7): 1393-1410.
57. Wang L, Hu W, Tan T. Recent Developments in Human Motion. Pattern Recognition. 2003; 36(3): 585-601.
58. Zhang T, Kuo CCJ. Audio Content Analysis for Online Audiovisual. IEEE Transactions on Speech and Audio. 2001; 9(4): 441-457.
59. Cybenko G. Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals, and Systems. 1989; 2(4): 303-314.
60. Ahmed MS. Neural-Net-Based Direct Adaptive Control for a Class of Nonlinear Plants. IEEE Transactions on Automatic Control. 2000; 45(1): 119-124.
61. Lightbody G, Irwin GW. Direct Neural Model Reference Adaptive Control. IEE Proceeding Control Theory Applications. 1995; 142(1): 31-43.
62. Limanond S, Si J. Neural-Network-Based Control Design: An LMI Approach. IEEE Transactions on Neural Networks. 1998; 9(6): 1422-1429.
63. Meireles MRG, Almeida PEM, Simões MG. A Comprehensive Review for Industrial Applicability of Artificial Neural Network. IEEE Transactions on ındustrial Electronics. 2003; 50(3): 585-601.
64. Noriega JR, Wang H. A direct adaptive neural-network control for unknown nonlinear systems and its application. IEEE Transactions on Neural Networks. 1998; 9(1): 27-34.
65. Omidvar O, Elliott DL. Neural Systems for Control: Elsevier; 1997.
66. Werbos P. An Overview of Neural Networks for Control. IEEE Control Systems. 1991; 11(1): 40-41.
67. Narendra KS. Intelligent Control. IEEE Control Systems. 19991; 11(1): 39-40.
68. Ku CC, Lee KY. Diagonal Recurrent Neural Networks for Dynamic Systems Control. IEEE Transactions on Neural Networks. 1995; 6(1): 144–156.
69. Narendra KS, Parthasarathy K. Identification and Control of Dynamical Systems Using Neural Networks. IEEE Transactions on Neural Networks. 1990; 1(1): 4-27.
70. Astrom KJ. Adaptive Feedback Control. Proceedings of the IEEE. 1987; 75(2): 185-217.

71. Astrom KJ,&WB. Adaptive Control Addison-Wesley , editor.: Longman Publishing Co.; 1994.
72. Middleton RH, Goodwin GC, Hill DJ, Mayne DQ. Design issues in adaptive control. IEEE Transactions on Automatic Control. 1988; 33(1): 50-58.
73. Narendra KS, Kudva P. Stable Adaptive Schemes for System Identification and Control Part-I. IEEE Transactions on Systems, Man and Cybernetics. 1974; 4(6): 542-551.
74. Sastry S, Bodson M. Adaptive Control Stability, Convergence and Robustness: Prentice Hall; 1989.
75. Pan Z, Başar T. Adaptive controller design for tracking and disturbance attenuation in parametric strict-feedback nonlinear systems. IEEE Transactions on Automatic Control. 1998; 43(8): 1066-1084.
76. Wu H, Hu J, Xie Y. Characteristic Model-Based All-Coefficient Adaptive Control Method and Its Applications. IEEE Transactions on Systems Man and Cybernetics Part C- Applications and Reviews. 2007; 37(2): 213-221.
77. Hatwell MS, Oderkerk BJ, Sacher CA, Inbar GF. The development of a model reference adaptive controller to control the knee joint of paraplegics. IEEE Transactions on Automatic Control. 1991; 36(6): 683-691.
78. Narendra KS, Balakrishnan J. Adaptive control using multiple models. IEEE Transactions on Automatic Control. 1997; 42(2): 171-187.
79. Gu K, Kharitonov VL, Chen J. Stability of Time-Delay Systems. 1st ed. New York: Springer-Verlag; 2003.
80. Vapnik VN. Statistical Learning Theory. 1st ed.: Wiley; 1998.
81. Cazzolato BS, Prime Z. On the Dynamics of the Furuta Pendulum. Journal of Control Science and Engineering. 2011 February;(3): 8 Pages.
82. Liu Y, Yu H. A survey of underactuated mechanical systems. IET Control Theory and Applications. 2013; 7(7): 921-935.
83. Nasır ANK. Modeling and controller design for an inverted pendulum system. Master's Thesis. Johor Bahru: University Technology Malaysia; 2007.
84. Subbotin MV. Balancing an Inverted Pendulum on Seesaw. Project Report. Santa Barbara: University of California; 2004.
85. Ljung L. System Identification: Theory for the User. 2nd ed.: Prentice Hall; 1999.
86. Ogata K. Modern Control Engineering: Pearson; 2009.
87. Wikipedia. [Online]. [cited 2019 December 21. Available from: https://www.wikizeroo.org/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvUElEX2NvbnRyb2xsZXI.

CURRICULUM VITAE

Mehmet Uğur Soydemir was born in May 1994, in Antalya, Turkey. He completed his undergraduate education in Ege University at Department of Electrical and Electronics Engineering in 2016. He completed his master degree in Izmir Katip Çelebi University at department of Electrical and Electronic Engineering in 2018. During his education, he worked on projects related and control systems fields.

List of Publications:

1. Soydemir MU, Alkuş İ, Bulucu P, Kocaoğlu A, Güzeliş C, Şahin S. Data Dependent Stable Robust Adaptive Controller Design for Altitude Control of Quadrotor Model. In 2018 18th International Conference on Mechatronics-Mechatronika (ME); 2018; Brno.
2. Bulucu P, Soydemir MU, Şahin S, Kocaoğlu A, Güzeliş C. Performance Analysis of Stable Adaptive NARMA Controller Scheme for Furuta Pendulum. In 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC); 2019; Sinaia.