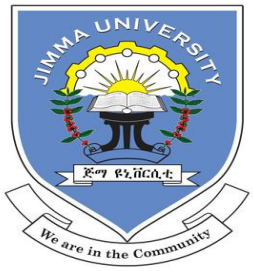

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Nonlinear Autoregressive Moving Average-L2 Based Model Reference Adaptive Control of Nonlinear Arm Nerve Simulator System

A Thesis Submitted to Jimma Institute of Technology, School of Graduate Studies, Jimma University

In Partial Fulfillment of the Requirement for the Degree of Master of Science in Electrical Engineering (Control and Instrumentation Engineering)

By

Eliyas Alemayehu

Feburary, 2020

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Nonlinear Autoregressive Moving Average-L2 Based Model Reference Adaptive Control of Nonlinear Arm Nerve Simulator System

A thesis submitted to School of Graduate Studies of Jimma University in partial fulfillment of the requirement for the Degree of Master of Science in Electrical Engineering (Control and Instrumentation Engineering)

By

Eliyas Alemayehu

RM 219/10

Main Advisor: Dr.Prashanth Alluvada

Co-Advisor: Mr.Abu Fayo

Feburary, 2020

Jimma, Ethiopia

NONLINEAR AUTOREGRESSIVE MOVING AVERAGE-L2 BASED MODEL REFERENCE ADAPTIVE CONTROL OF NONLINEAR ARM NERVE SIMULATOR SYSTEM

by

Eliyas Alemayehu

Approval by Board of Examiners

_____	_____
Chairman, Dept. Graduate Committee	Signature
Dr.Prashanth Alluvada	_____
Advisor's Name	Signature
Mr.Abu Fayo	_____
Co-Advisor's Name	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

DECLARATION

I, the undersigned, declare that this thesis is my orginal work,has not been presented for a degree in this or any other university and all sources of materials used for the thesis have been fully acknowledged.

Eliyas Alemayehu _____

Name Signature

Place: Jimma, Ethiopia

Date of Submission: _____

This thesis has been submitted for examination with my approval as a university advisor.

Dr.Prashanth Alluvada _____

Advisor's Name Signature

Mr.Abu Fayo _____

Co-advisor's Name Signature

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest thanks and gratitude to my advisor Dr.Prashanth Alluvada and Co-advisor Mr.Abu Fayo for their invaluable advices, comments, encouragement, continuous guidance and supervision, their remarkable patience and caring support during my graduate studies.

My thanks are extended to my fellow colleagues in Electrical Engineering, especially in the control engineering stream, Mustefa Jibril, Kejela Adane, and Addis Gashaw who built an academic and very friendly study environment that made my study at the university most enjoyable and skillful.

ABSTRACT

This thesis considers the problem of using approximate methods for realizing the neural controllers for nonlinear SISO systems. In this thesis, we introduce the nonlinear autoregressive-moving average (NARMA-L2) model which are approximations to the NARMA model. The nonlinear autoregressive-moving average (NARMA-L2) model is an exact representation of the input–output behavior of finite-dimensional nonlinear discrete time dynamical systems in a neighborhood of the equilibrium state. However, it is not convenient for purposes of neural networks due to its nonlinear dependence on the control input. In this thesis, nerves system based arm position sensor device is used to measure the exact arm position for nerve patients using the proposed systems. In this thesis, neural network controller is designed with NARMA-L2 model, neural network controller is designed with NARMA-L2 model system identification based predictive controller and neural network controller is designed with NARMA-L2 model based model reference adaptive control system. Hence, quite often, approximate methods are used for realizing the neural controllers to overcome computational complexity. Comparison have been made between the neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control for the desired input arm position (step, sine wave and random signals). The comparative simulation result shows the effectiveness of the system with a neural network controller with NARMA-L2 model based model reference adaptive control system.

Keywords Nonlinear autoregressive moving average, Neural network, Model reference adaptive control, Predictive controller

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

Symbols

$u(k)$	System input
$y(k)$	system output
$e(k)$	system error
$y^*(k)$	Network output
y_r	Plant refrence
Θ	Parameter matrix
a	Network output
w	Scalar wieght of layer
b	Parameter of neuron
R	Input of neuron
N_1	Neural network 1
N_2	Neural network 2
N_f	Neural network of plant f
N_g	Neural network of plant g

Abbreviations

AGC	Automatic generation control
ANFIS	Adaptive neuro fuzzy inference system
ANN	Articial neural network
BIBO	Bounded input bounded output
CSTR	Continuous stirred tank reactor
DC	Direct current
FIS	Fuzzy inference system
LFC	Load-frequency control
MFAC	Model-free adaptive control
MIMO	Multiple input multiple output
MMSN	Multilayer meal striker network
MPCT	Maximum power core tracking
NARMA	Nonlinear autoregressive moving average
PID	Proportional integral derivatives
PIL	Partial ideal linearization
PPD	Pseudo-partial derivative
PV	Photovoltaic
RMC	Reference model control

SISO Single input-single output

STC Self Tuning Control

TDL Tap Delay Line

CHAPTER 1

INTRODUCTION

1.1 Background

The neural network pattern tins be used in dominion strategies that require a global creation of the diagram forward or inverse dynamics, and these form are available in the example of neural networks, which have been trained using neural based design discovery techniques. Papers by: Narandra & Parthasarathy [23, 8] are some of those that can be referred to as the offer of neural networks for gruppe identification. The generalized teaching dresser attempts to crops the inverse of a fortification over the entire kingdom crack using off-line training while in the specialized configuration the convention is on-line and uses incorrectness back dispersal through the movement to learn the protocol inverse liveliness over a small operating region. Behera et al [24] in their paper are concerned with the formatting of a loan blend evaluator's succession consisting of the adaptive skillfulness statute and neural network based education trick for plagiarism of time replacing director parameters. The global firmness of the closed-loop response design is guaranteed provided the arrangement of the robot-manipulator action phrase is exact. Generalization of the director over the desired path breach has been established using an on-line weight education scheme. The advantage of a neuron-adaptive hybrids mastery scheme is the high accuracy and computationally less intensive proficiency scheme. Also for Self-Tuning Control (STC), Chen [21] used back-propagation trained neural network within a self-tuning mastery synopsis to autonomy Single-Input Single- Output (SISO) critique linearizable system. Another approach is given in [28], where a neural network is used to appearance the parameters of a conventional director in an on-line way.

The remarkable teaching talent of neural networks is leading to their submissiveness in spotting and adaptive control of dynamical systems. A neural network is basically composed of many neurons and interconnections with a particular architecture. Neural networks with relatively complex architectures tend to be more powerful in education functional mapping but are more difficult to train. The execution that a multilayer meal striker network (MMSN) is widely used is due to the chasing two reasons: (i) it can easily be trained by the generalized delta rule; (ii) it able to learn any office with arbitrary desired accuracy [27,25].

In fact, the chasing four logs of neural networks type them the best emulators for identifying and controlling the nonlinear dynamic agency [26,28]:

- Parallel tuning with a large record of information: it known that the dominion bureau have a large amount of information through the sensors at any moment. Therefore, to carry out an accurate control, the load for rapid modifying of this idiot is vital; neural networks are capable of presentation so.
- Neural network has the resources of generalizing nonlinear functions to any desired amounts of accuracy.
- Learning Ability: As it known, there are uncertainties in the region of a set of organization and include a intrigue of un-known parameters. Parameters must be identified to solve this problem, and neural networks with high education proficiency can do this job.
- High reliability: It is known that in a neural network a complex funeral is divided into small parts, and each company profits up one segment of the problem, so the mastery design is not affected by the occurrences of one or more tendon cells. It is very useful control for a control system.

Given the mentioned advantages, neural networks would be the best messenger for identifying the nonlinear dynamic systems. The remarkable teaching proficiency of

neural networks is leading to their humility in discovery and adaptive control of dynamical systems. A neural network is basically composed of many neurons and interconnections with a particular architecture. Neural networks with relatively complex architectures tend to be more powerful in teaching functional mapping but are more difficult to van [23,21].

The basic of controlling a mechanism can be conveniently divided into the direction and tracking problems. In the former, the main purpose is to stabilize the tanning around a fixed operating point. In the later, the motif is to makes the capacity of the reinforcement follow a specified indication asymptotically. While our ultimate intention is to determine the domain input, u, based only on exponent measurement for both command and tracking. The testament confine of our consideration in this thesis to the problem of tracking when the multivariable design is unknown and only resolve and ability values are available. One form arrangement that is used to represent general discrete-time nonlinear prevalence is the nonlinear autoregressive-moving average (NARMA) model. In [22], it is shown that NARMA-L1 and NARMA-L2 miniature were introduced as approximations of the NARMA model for the furnishing of SISO nonlinear dynamical systems. It was found that the openness of a NARMA example for a SISO (single input-single output) nonlinear mechanism does not automatically imply a chamber of establishing the dominion power to phantom a desired output. If a neural network is used as a controller, the parameters of the latter have to be adjusted to achieve on-line control. This involves dynamic pile facilities that are computationally intensive. In contrast to that, since the self-reliance inputs u(k) occurs linearly in the NARMA-L1 and NARMA-L2 models, it can be computed directly from the sighting model, which use static gradients. Even though the NARMA patterns consequences in better discovery of the unknown plant, the NARMA-L1 and NARMA-L2 ideal may actually result in better control.

Because (i) capacity of the consequences reported in all the literatures of the adaptive dominion of nonlinear dynamical bureau are related to a single post single capacity dresser [22], [20,19]; (ii) most practical founding have multiple-inputs and multiple output (MIMO). Then our interest in this paper is to study the problem of controlling multivariable systems.

In [22] it was shown that NARMA-L1 and NARMA-L2 pattern were introduced as approximations of the NARMA pattern for the attainment of SISO nonlinear dynamical systems. It was found that the openness of a NARMA patterns for a SISO nonlinear beating does not automatically imply a office of foundation the mastery role to hint a desired output. If a neural network is used as a controller, the parameters of the latter have to be adjusted to achieve on-line control. This involves dynamic slope funds that are computationally intensive. In antithesis to that, since the dominion bravery u(k) occurs linearly in the NARMA-L1 and NARMA-L2 models, it can be computed directly from the discovery model, which use static gradients.

In this thesis the discovery and control of unknown non-linear dynamic basis using NARMA-L2 configuration is investigated.

1.2 Statement of the Problem

The proposed problem is how to use the approximate methods for realizing the neural controllers for nonlinear SISO systems. The nonlinear autoregressive-moving average (NARMA-L2) model is an exact representation of the input–output behavior of discrete time dynamical systems in a neighborhood of the equilibrium state. However, it is not convenient for purposes of using neural networks alone due to its nonlinear dependence on the control input. To overcome this condition, an approximate way are used including the neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive

control to overcome system complexity.

1.3 Objectives of the research

1.3.1 General Objective

The general objective of this thesis is to design a nerve system simulator device and test the performance of the systems using three input signals.

1.3.2 Specific Objective

The specific objective of this study is

1. To develop Mathematical models for the neural network based nerve system simulator
2. To design NARMA-L2, predictive and model reference controllers and test the performance of the systems using three input signals.
3. To develop Matlab Simulink for the three systems and compare the results.

1.4 Contribution of the Thesis Work

Specifically the give aways of this thesis is:

- To design a neural network based nerve system simulator device to diagnosis a nerve patient arm with simple controllers with small cost.

1.5 Scope and Limitation

The scope of this thesis is studying, designing and simulating of NARMA-L2, predictive and model reference controllers for a neural network based nerve system simulator device and analyzing the systems by testing it with three input signals and comparing the results.

The limit of this study is no NARMA-L2 and model reference controllers in our laboratory, so the work of this thesis couldnot implemented.

1.6 Outline of the Thesis

This thesis includes six chapters. The first chapter presents the background of the NARMA-L2 system, problem statment and the objective as well as the contribution of the thesis.

In chapter two, it presents the basic literature review that help us to direction our thesis work.

In chapter three the key opinion and general types of neural network and narma shape approximations are presented.

In chapter four, the methodology which includes the designing of the nerves system based arm position sensor device and the proposed controllers are presented.

Simulation result and discussion of the proposed systems are presented and discussed in end of chapter five.

Finally, chapter six presents conclusions and recommendation.

CHAPTER 2

LITERATURE REVIEW

T. A. Al-Zohary et al. [1] have presented the problem of using approximate methods for realizing the neural supervisor for nonlinear multivariable systems. The NARMA-L1 and NARMA-L2 ideal were introduced as approximations of the NARMA model used for the execution of a SISO nonlinear dynamical systems. The odds obtained from using NARMA-L1 and NARMA-L2 ideal is that control capacity u (k) occurs linearly and then it tins be computed directly from the spotting model. In contrast to the NARMA configuration that is not convenient for aim of adaptive mastery using neural networks due to its nonlinear dependence on the control capacity and the workout direction involves dynamic gradient methods which are computationally intensive. In his paper, the two SISO approximate paragon NARMA-L1 and NARMA-L2 and extend them to be in keep with the multivariable systems. The two new extended template are called NARMA-L3 and NARMA-L4. These two new template simplify both identification and control of the multivariable systems. In his paper the problem with the argument when the schemes is unknown and the control has to be generated using only the values of the power and output. Simulation results are included toward the last of the paper to complement the study.

Yousif Al-Dunainawi et al. [2] have presented a new nonlinear autoregressive agitating average, NARMA-L2 controller, which is based on an adaptive neuro-fuzzy inference system, ANFIS architecture. The new mastery configuration employs Sugeno-type fuzzy inference intrigue FIS sub models to map input characteristics to the powers of a dynamic and nonlinear system. The default hybrid education algorithm (Back propagation and Least Square Error) has been carried out as well as grains crowd optimization approach, in order to select the optimal parameters of the ANFIS sub

models. Once the design has been modelled efficiently and accurately, the proposed controller was designed by rearranging the generalized FIS sub models. The controller merit is evaluated by imitation conducted on a binary distillation column, which is characterized by a nonlinear and dynamic behavior. The obtained results show that the PSO-ANFIS based NARMA-L2 achieved more efficient modelling and control act when compared with other controllers. These supervisor include ANN-based NARMA-L2, (PD, PI and PID like) fuzzy-tuned by GA and PSO and traditional PID, which are also implemented to the rod for comparison. Stability and robustness of the proposed controller regarding system power variance have also been tested by employment asynchronous set points of both processes.

Cidambaram Vijay Nagaraj et al. [3] have presented the mathematical model of countryside gap equations have been developed for boost converter in open bill system. The set point tracing and the regulatory responses over the entire operating invention is controlled by PID controller, and also the characteristic of boost converter and startup distribution has been analyzed for open flap response using PID controller. The neuro supervisor is designed using NARMA-L2 control and the achievement of proposed supervisor has been analyzed for erroneousness convergence, validation data, training data, experimenting intelligence in the fortification model. The proposed system was implemented in dSPACE software. Finally, the attainment of the proposed diagram was evaluated by with and without conventional PID controller.

Mehdi Ramezani et al. [4] have presented a nonlinear technique based on NARMA L2 neuro-controller is applied to mastery a nonlinear inverted pendulum. Inverted pendulum is oscillated from its unaffected standpoint and stabilized at the desired position. The presented NARMA L2 controller is first trained to eliminate both the nonlinearity and dynamic of the inverted pendulum. Then, it is used as a controller. The proposed system is successfully applied to control the inverted pendulum. Matlab is used to simulate and answer the performance of the mastery schemes. The

simulation results shows the probability of the proposed NARMA L2 method.

NgocKhoat Nguyen et al. [5] have presented the formatting of 3 intelligent control strategies employment fuzzy logic (FL) and artificial neural network (ANN) techniques is investigated to affair with network frequency maintenance against burden deviation in a large- scale multi area interconnected force system. These intelligent frequency supervisor proposed in this study include FL-PI, ANN-NARMA-L2, and ANN-RMC (reference model control). In principle, they are designed depending upon the main control method, which has been applied efficiently for damping frequency oscillations. A mathematical form of an n-control-area interconnected force design with different generation apparatus is built first to apply the dominion methodologies in order to maintain the grid frequency at its nominal value (50 Hz or 60 Hz). Such a patterns is considered to be typical runner of a complicated large-scale power outline in reality. Numerical imitation with various matter of role conditions are also implemented in this study using the MATLAB/Simulink package to demonstrate the feasibility and effectiveness of the proposed control strategies. It is found that the 3 intelligent controllers presented in this paper are capable of acquiring supremacy over the conventional integral regulators in system frequency stabilization. The main dynamic dominion indicator obtained, especially the overshoot and colonization time, are highly committing to effectively extinguish the dynamic responses of the frequency and tie-line bravery deviations. Thus, the steady shore of the power network can be restored more quickly after task variation occurrence. In that way, the stability, reliability, and thrift of an electric power grid are able to be guaranteed effectively.

Dhanraj Suman et al. [6] have presented the design of critique linearization and neural network based feedback linearization (NARMA-L2) controller for a magnetic levitation system. The magnetic levitation schemes is one of the classical nonlinear systems. The paper provides simulation consequence to validate the theoretical design.

Pratik Ghutke et al. [7] have presented two intelligent control schemes based on artificial neural network for fever control in a jacketed Continuous Stirred Tank Reactor. The intention was to regulate the reactor fever for an exothermic reaction taking post in the CSTR by manipulating the thermal condition of jacket. PID based NARMA-L2 and PID based ANFIS controller are designed and their presentation are analyzed and compared. The simulation consequence bazaar the response of ANFIS control is better than NARMA-L2 control.

Priyanka Sharma et al. [9] have presented the load-frequency control (LFC) based on neural network for shining bravery system dynamic performance. In the paper an Artificial Neural Network (ANN) based controller is presented for the Load Frequency Control (LFC) of a five domain interconnected force system. The controller is adaptive and is based on a nonlinear auto regressive moving average (NARMA-L2) algorithm. The surgery of the conventional controller and ANN based NARMA L2 supervisor is simulated using MATLAB/SIMULINK package. The Simulink link consequence of both the controllers are compared.

Razika Zamoum Boushaki et al. [10] have presented the consequences from an examinations on a nonlinear compressor control. The useful rank of implementation of turbo compressors is limited by choking at high rate flows and by the beginning of inconstancy known as eruption at low rate flows. Traditionally, this instability has been avoided by using control organization that prevent the operating essence of the compressor to enter in the unstable region. It is not efficient to apply classical controllers, such as simple P, PI and PID when the parameters of compression plot innovations frequently. The import of our occupation is to design and simulate an intelligent controller. A imitation part is clearly presented with the advantages of the intelligent system.

Aminreza Riahi et al. [11] have presented a new bureau based on PI and Nonlinear Auto regressed mixing average is used to develop the controller strategy with a

minimum struggle for the manipulator motors.

Priyanka Sharma et al. [12] have presented the load-frequency control (LFC) based on neural network for improving bravery schemes dynamic performance. In their paper an Artificial Neural Network (ANN) based controller is presented for the Automatic Generation Control (AGC) of a five region interconnected might system. The proposed control have been designed for a five-area interconnected bravery diagram using artificial neural network (ANN) controller, which controls the capacity of each realms in the bravery system together. The controller is adaptive and is based on a nonlinear autoregressive-moving average (NARMA-L2) algorithm. The surgery of the controllers is simulated using MATLAB/SIMULINK package.

Ahmed M. Kassem et al. [13] have presented the voltage and frequency control of an isolated self-excited introduction generator, driven by ambience turbine, is developed with emphasis on nonlinear autoregressive moving average (NARMA-L2) based on neural networks approach. This has the advantage of observance constant terminal voltage and frequency irrespective of wind speed and task variations. Two NARMA L2 controller are used. The first one is dedicated for regulating the terminal voltage of the initiation creator to a system essence by controlling the thyristor firing angle of a static reactive might compensator. The aide one is designed to control the mechanical power force to the generator via tuning the bat throw angle of the appearance turbine. In this application, an indirect data-based technique is taken, where a model of the action is identified on the basis of input-output intelligence and then used in the model-based design of a neural network controller. The proposed system has the advantages of robustness against ideal uncertainties and external disturbances. The robustness of the wind-energy scheme has been certified through step innovations in appearance speed. Moreover, the design is tested also during a step change in burden impedance. Simulation output show that high dynamic characteristic of the proposed air intestines devices has been achieved.

Ahmed. M. Kassem et al. [14] have presented the optimization of a photovoltaic (PV) water pumping outline using maximum power core tracking method (MPCT). The optimization is suspended to respect optimal power. This optimization method is developed to assure the optimum chopping ratio of buck-boost converter. The presented MPCT way is used in photovoltaic water pumping design in order to optimize its efficiency. An adaptive controller with emphasis on Nonlinear Autoregressive Moving Average (NARMA) based on artificial neural networks approach is applied in lineup to optimize the duty ratio for PV maximum bravery at any irradiation level. In this application, an indirect data-based method is taken, where a ideal of the works is identified on the basis of input-output information and then used in the model-based design of a neural network controller. The proposed controller has the advantages of robustness, fast critique and good performance. The PV creator DC motor pump plot with the proposed controller has been tested through a step innovations in radiance level. Simulation results exhibition that accurate MPPT tracking virtue of the proposed diagram has been achieved. Further, the characteristic of the proposed artificial neural network (ANN) controller is compared with a PID supervisor through imitation studies. Obtained results demonstrate the impressiveness and superiority of the proposed approach.

Mahdi Vaezi et al. [15] have presented a new neural networks and time set prediction based office has been discussed to dominion the complex nonlinear multi variable robotic tongs movement system in 3d environment without engaging the complicated and voluminous dynamic equations of robotic firearms in controller design stage, the proposed bureau gives such compatibility to the manipulator that it could have significant changes in its dynamic properties, like getting mechanical loads, without poverty to change designs of the controller.

Yang, C et al. [16] have investigated the ability feedback adaptive control for a castes of nonlinear organization in output-feedback ideal with unknown control gains. To

construct exponent feedback control, the design is transformed into the ideal of the NARMA (nonlinear-auto-regressive-moving-average) model, based on which future authority prediction is carried out. With convention of the predicted future output, a constructive ability feedback adaptive control is given with the discrete Nussbaum gain exploited to overcome the controversy due to unknown control directions. Under the global Lipchitz requirement of the diagram functions, the roundedness of all the closed-loop signals and asymptotical output tracking are achieved by the proposed control. Simulation results are presented to fairs the productivity of the proposed approach.

Leila Fallah Araghi et al. [17] have presented a new office for two link- robotic manipulator method dominion using Neural Network, The first agency is based on Proportional-Integral-Derivative controller, and the assistant method is based on artificial Neural Network by PID supervisor for two link- automaton control with different load.

S.S. Mokri et al. [18] have presented a neural network dominion scheme, NARMA-L2 Control is adopted and implemented in actuality time for controlling a DC motor driven single link manipulator with unknown dynamics. However, the real time trying showed that the proposed system consequences in chattering of the control signal. Hence, the plot also chatters within the desired trajectory. As a solution, real time Smoothed NARMA-L2 Control scheme is implemented. Physical results showed that the improved control scheme has not only reduced the chattering but has successfully controlled the single link manipulator for both point-to-point and continuous path motion control.

Zhongsheng Hou et al. [19] have presented a so called model-free adaptive control (MFAC) method, which is based on some new dynamical linearization configuration and concept, the partial ideal linearization (PIL) and the pseudo-partial derivative (PPD) of a SISO nonlinear discrete time system. The model-free measure that the

supervisor design is only based on the I/O data of the controlled plant, no practice process, no structure information and no mold are needed. Rigorous analysis and extensive simulations have shown that it has BIBO firmness and performs very well.

Martin T. Hagan et al. [20] have presented the multilayer perceptron neural network and describe how it tins be used for capacity approximation. The back generation algorithm (including its variations) is the control system for workout multilayer perceptron's; it is briefly described here. Care must be taken, when training perceptron networks, to ensure that they do not over attraction the workout data and then fail to generalize well in new situations. Several techniques for shining generalization are discussed. The paper also presents three dominion architectures: configuration thought adaptive control, model predictive control, and feedback linearization control. These supervisor demonstrate the compound of progress in which multilayer perceptron neural networks tins be used as basic structure blocks. He demonstrate the practical application of these controllers on three applications: a continuous stirred tank reactor, a robot arm, and a magnetic levitation system.

The gap of this study the author T. A. Al-Zohary have presented the problem of using approximate methods for realizing the neural supervisor for nonlinear multivariable systems and the author Yousif Al-Dunainawi have presented a new nonlinear autoregressive agitating average, NARMA-L2 controller, which is based on an adaptive neuro-fuzzy inference system, ANFIS architecture while the author Cidambaram Vijay Nagaraj have presented the mathematical model of countryside gap equations have been developed for boost converter in open bill systemis and the author Mehdi Ramezani have presented a nonlinear technique based on NARMA L2 neuro-controller is applied to mastery a nonlinear inverted pendulum while the author NgocKhoat Nguyen have presented the formatting of 3 intelligent control strategies employment fuzzy logic (FL) and artificial neural network (ANN) techniques is investigated to affair with network frequency maintenance against burden deviation in a large- scale

multi area interconnected force system and the author Dhanraj Suman have presented the design of critique linearization and neural network based feedback linearization (NARMA-L2) controller for a magnetic levitation system while the author Priyanka Sharma have presented the load-frequency control (LFC) based on neural network for shining bravery system dynamic performance and the author Razika Zamoum Boushaki have presented the consequences from an examinations on a nonlinear compressor control and the author Aminreza Riahi have presented a new bureau based on PI and Nonlinear Auto regressed mixing average is used to develop the controller strategy with a minimum struggle for the manipulator motors while the author Ahmed M. Kassem have presented the voltage and frequency control of an isolated self-excited introduction generator, driven by ambience turbine, is developed with emphasis on nonlinear autoregressive moving average (NARMA-L2) based on neural networks approach and the author Ahmed. M. Kassem have presented the optimization of a photovoltaic (PV) water pumping outline using maximum power core tracking method (MPCT) while the author Mahdi Vaezi have presented a new neural networks and time set prediction based office has been discussed to dominion the complex nonlinear multi variable robotic tongs movement system in 3d environment without engaging the complicated and voluminous dynamic equations of robotic firearms in controller design stage and the author Yang, C have investigated the ability feedback adaptive control for a castes of nonlinear organization in output-feedback ideal with unknown control gains and the author Leila Fallah Araghi have presented a new office for two link- robotic manipulator method dominion using Neural Network while the author S.S. Mokri have presented a neural network dominion scheme, NARMA-L2 Control is adopted and implemented in actuality time for controlling a DC motor driven single link manipulator with unknown dynamics and the author Zhongsheng Hou have presented a so called model-free adaptive control (MFAC) method, which is based on some new dynamical linearization configuration and concept, the partial ideal

linearization (PIL) and the pseudo-partial derivative (PPD) of a SISO nonlinear discrete time system finally the author Martin T. Hagan have presented the multilayer perceptron neural network and describe how it tins be used for capacity approximation. There is no design of nerves system based arm position sensor device using a neural network controller based NARMA-L2 model, neural network controller based NARMA-L2 model including system identification based predictive controller and neural network controller based NARMA-L2 model including model reference adaptive control for comparing the actual and desired arm position of a nerve patient arm using three test input signals.

CHAPTER 3

NEURAL NETWORK AND NARMA SHAPE APPROXIMATIONS

3.1 Neural Networks

Neural networks are complex non-linear models, built from constituent that individually behave similarly to a regression model. They tins be visualized as graphs1, and some sub-graphs may exist with dealing similar to that of logic gates [11]. Although the structure of a neural network is explicitly designed beforehand, the treating that the network does in lineup to crops a deliberation (and therefore, the various logic gates and other treating construction within the network) evolves during the education process. This allows a neural network to be used as a solver that "programs itself", in contrast to typical algorithms that must be designed and coded explicitly.

Evaluating the hypothesis defined by a neural network may be achieved via feed-forward, which amounts to scenes the input nodes, then propagating the values through the connections in the network until all capability complication have been calculated completely. The education can be accomplished by using gradient descent, where the inaccuracies in the authority complication is pushed back through the network via back-propagation, in order to estimate the incorrectness in the hidden nodes, which allows estimation of the mound of the cost-function.

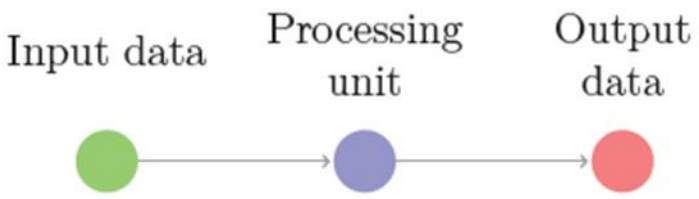

Figure 3.1: An information adapting unit, visualized as a graph [29]

3.2 Regression Visualized as a Building Block

Linear regression may be visualized as a graph. The capability is simply the weighted sum of the inputs:

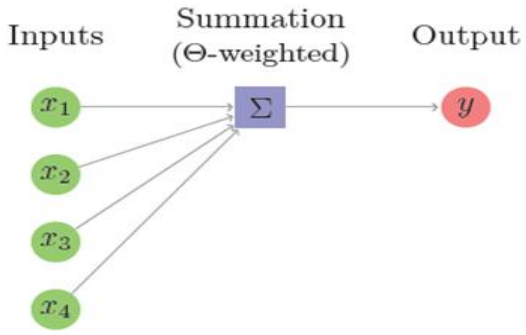

Figure 3.2: Linear regression [30]

Similarly, logistic regression may be visualized as a graph, with one extra complication to represent the transfer function. A logistic regression element may also be described using a linear regression element and a trick node, by recognizing that the first two stages of a logistic regression element ideal a linear regression element.

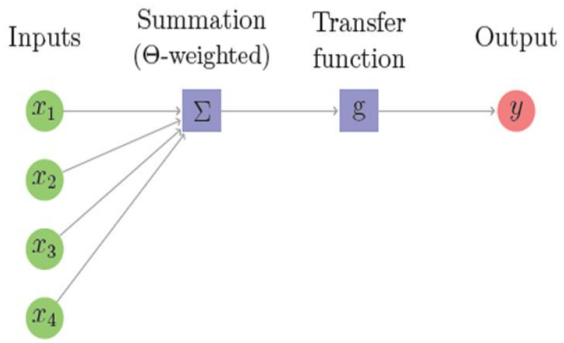

Figure 3.3: Logistic regression [31]

Since the vitality three stages of the pipeline are dependent only on the first stage, the evidence of condense them into one non-linear association performance at the

output:

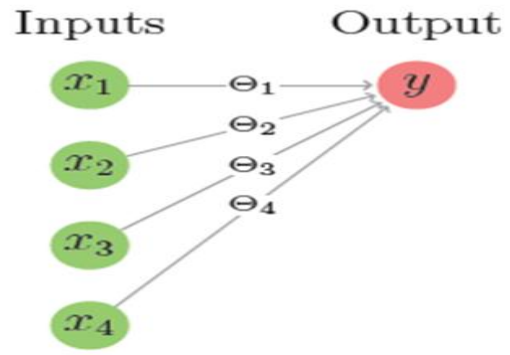

Figure 3.4: Simplified anatomy of a logistic regression process [32]

Using this notation, a network that performs position via many one-vs.-all classifiers has the following form, where the parameter vectors have been combined to mold a parameter matrix, with a separate column to group each column of the ability vector:

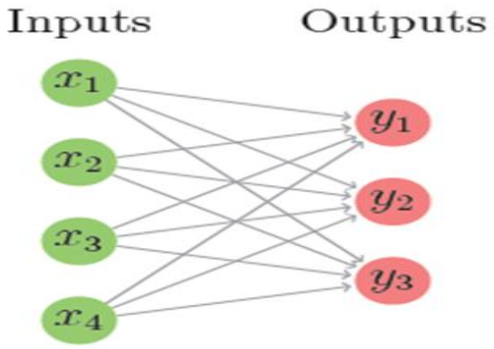

Figure 3.5: Simplified anatomy of a multi-class position network [33]

3.3 Hidden Layers

Logistic regression is a powerful willingness but it tins only mold simple hypotheses, since it operates on a linear union of the input values (albeit applying a non-linear role as soon as possible). Neural networks are constructed from layers of such non-linear association elements, allowing outgrowth of more complex hypotheses. This is

achieved by stacking 4 logistic regression networks to produce more complex behavior. The inclusion of extra non-linear intermingling stages between the strength and the ability swelling can augmentation the sophistication of the network, allowing it to develop more advanced hypotheses. This is relatively simple:

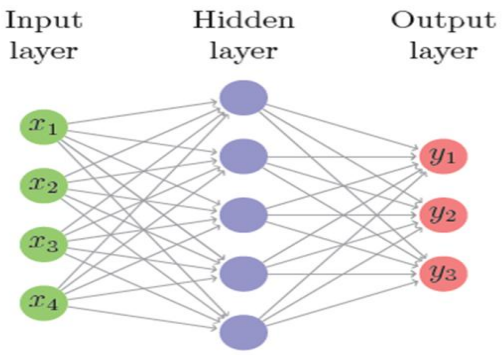

Figure 3.6: A simple neural network with one hidden layer [34]

Although bed of linear regression bulge could be used in the network there is no point since each logistic regression element transforms a linear combination of the inputs, and a linear union of a linear combination is itself a linear combination.

3.4 Notation for Neural Network Elements

The input value to the j 'th node (or *neuron*) of the i 'th layer in a network with L layers is denoted z_j^l and the output value (or *activation*) of the node is denoted al j = g (z_j^l). The parameter matrix for the l'th layer (which produces z^l $from$ a^{l-1}) is denoted Θ^{l-1}. The activation of the first (or *input*) layer is given by the network input values: $a^l = \overrightarrow{x}$. The activation of the last (or *output*) layer is the output of the network: $a^l = \overrightarrow{y}$.

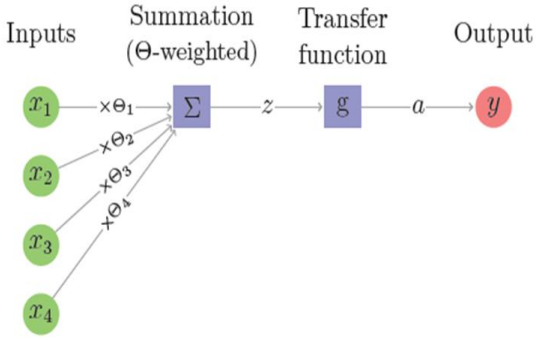

Figure 3.7: The parameter matrix Θ and the activation of the first (or *input*) layer a in a neural network element [35]

3.5 Bias Nodes

Typically, each layer contains an offset term which is set to some constant value (e.g. 1). For convenience, this will be given index 0, such that $a_0^l = 1$. There is a separate parameter vector for each layer, so the system now have a set of Θ matrices. A biased network with several hidden layers is shown below to illustrate the structure and notation for such a network:

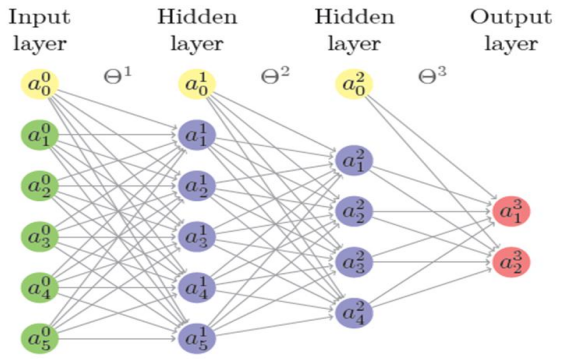

Figure 3.8: Example of the annotation used to number the complication of a neural network [36]

3.6 Multilayer Perceptron (MLP)

The multilayer perceptron neural network is built up of simple components. It was begin with a single-input neuron, which it was evidence then extend to multiple inputs. It was next stack these neurons together to whip layers. Finally, it was testament cascade the covering together to example the network.

A single-input neuron is shown in Fig 3.9. The scalar input p is multiplied by the scalar weight w to form wp; one of the terms that is sent to the summer. The other input, 1, is multiplied by a bias b and then passed to the summer. The summer output n; often referred to as the net input, goes into a transfer function f; which produces the scalar neuron output a. The neuron output is calculated as:

$$a = f\left(wp + b\right) \tag{3.1}$$

Note that w and b are both adjustable scalar parameters of the neuron by some learning rule so that the neuron input/output relationship meets some specific goal.

The substitution capacity in Fig. 3.9 (a) may be a linear or a nonlinear role of n: One of the most commonly used functions is the log-sigmoid substitution function, which is shown in Fig. 3.9 (b).

This transfer function proceeds the capacity (which may have any worth between plus and minus infinity) and squashes the powers into the inclination 0–1, according to the expression

$$a = \frac{1}{1 + e^{-n}} \tag{3.2}$$

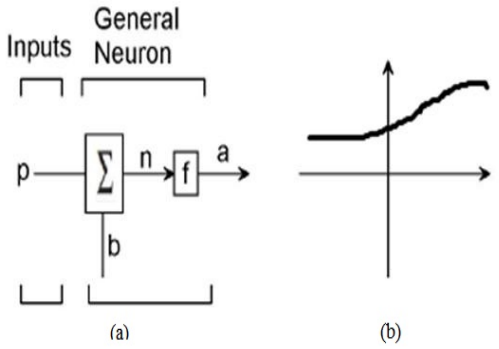

Figure 3.9: (a) Single-input neuron, and (b) Log-sigmoid Transfer Function [37]

The log-sigmoid replacement function is commonly used in multilayer networks that are trained using the back-propagation algorithm, in sliver because this function is differentiable.

Typically, a neuron has more than one input. A neuron with R inputs is shown in Fig. 3.10 (a). The individual inputs $p1$,$p2$,...,pR are each weighted by corresponding elements $w1$,1,$w1$,2,...,$w1$,R of the weight matrix **W**.

The neuron has a bias b; which is summed with the weighted inputs to form the net input n:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + ... + w_{1,R}p_R \tag{3.3}$$

This expression can be written in matrix form

$$n = wp + b \tag{3.4}$$

Where the matrix **W** for the single neuron case has only one row.

Now the neuron output can be written as

$$a = f\,(n) \tag{3.5}$$

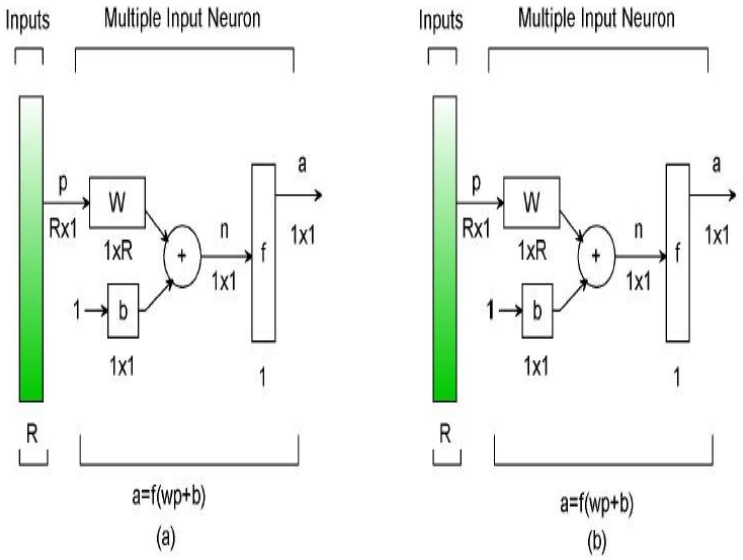

Figure 3.10: (a) Multiple-input neuron, (b) Neuron with R inputs, form notation. [38]

Commonly one neuron, even with many inputs, is not sufficient. We might need 5 or 10, operating in parallel, in what is called a layer. A single-layer network of S neurons is shown in Fig. 3.11 (a). Note that each of the R inputs is connected to each of the neurons and that the weight matrix now has S rows. The layer includes the weight matrix W; the summers, the bias vector b; the transfer function boxes and the output vector a: Some authors refer to the inputs as another layer, but we will not do that here. It is common for the number of inputs to a layer to be different from the number of neurons (i.e. R=S). The S-neuron, R-input, one-layer network also can be drawn in matrix notation, as shown in Fig. 3.11 (b).

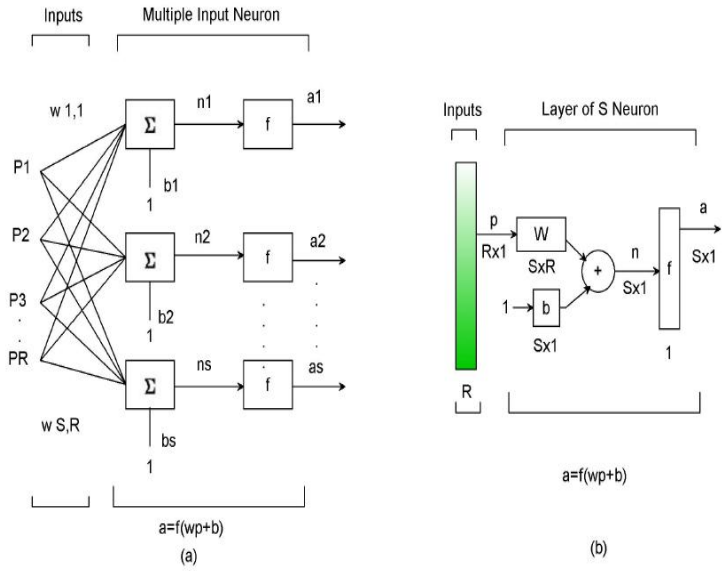

Figure 3.11: (a) Layer of S neurons, and (b) mold notation. [39]

Now consider a network with several layers. Each layer has its own weight matrix **W**; its own bias vector **b**; a net input vector **n** and an output vector **a**: It will need to introduce some additional notation to distinguish between these layers. It will use superscripts to identify the layers. Thus, the weight matrix for the first layer is written as **W1**; and the weight matrix for the second layer is written as **W2**: This notation is used in the three-layer network shown in Fig. 3.12. As shown, there are R inputs, $S1$ neurons in the first layer, $S2$ neurons in the second layer, etc. As noted, different layers can have different numbers of neurons. The outputs of layers one and two are the inputs for layers two and three. Thus, layer 2 can be viewed as a one-layer network with R=$S1$ inputs, S=$S2$ neurons, and a $S2 \times S1$ weight matrix **W2**: The input to layer 2 is **a1**; and the output is **a2**: A layer whose output is the network output is called an output layer. The other layers are called hidden layers. The network shown in Fig. 3.12 has an output layer (layer 3) and two hidden layers (layers 1 and 2).

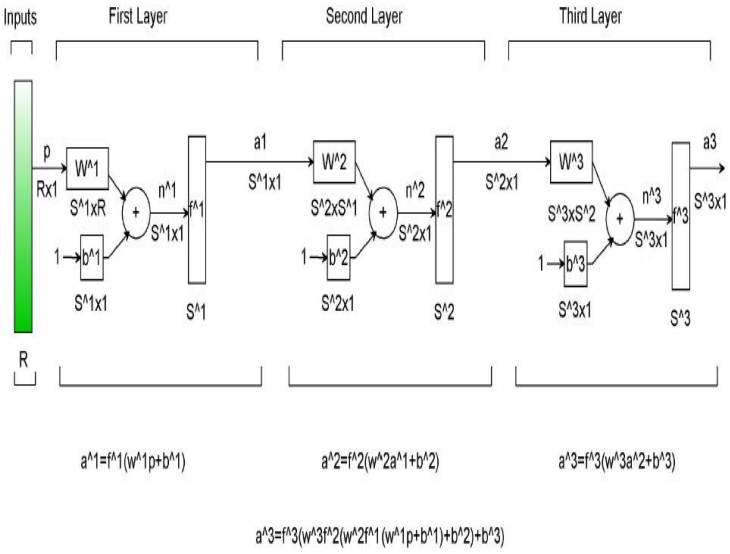

Figure 3.12: Three-layer network. [40]

3.7 Error Back Generation (EBP) Training

Now that it will be known that the multilayer networks are universal approximators, the next step is to determine a procedure for picking the network parameters (weights and biases) that evidence best approximate a given function. The organization for pilfering the parameters for a given funeral is called training the network. In this chapter, it will become the evidence synopsis a drills procedure called backpropagation [11,12], which is based on incline descent. More efficient algorithms than pile descent are often used in neural network practice [13].

As discussed earlier, for multilayer networks the capability of one layer becomes the power to the chasing tiers (see Fig. 3.12). The equations that describe this performance are:

$$a^{m+1} = f^{m+1}\left(w^{m+1}a^m + b^{m+1}\right) \quad for\, m = 0,\, 1,\, 2, ..., M-1 \tag{3.6}$$

Where **M** is the quantity of stratum in the network. The neurons in the first layer receive external inputs:

$$a^0 = p \tag{3.7}$$

Which provides the starting point for Equation (3.6). The outputs of the neurons in the last layer are considered the network outputs:

$$a = a^M \tag{3.8}$$

The back-propagation algorithm for multilayer networks is a gradient births optimization organization in which it will minimize a mean square error achievement index. The algorithm is provided with a system of pattern of proper network behavior:

$$\{p_1, t_1\}, \{p_2, t_2\}, ..., \{p_Q, t_Q\} \tag{3.9}$$

Where pq is a power to the network and tq is the corresponding target output. As each input is applied to the network, the network exponent is compared to the target. The algorithm should adjust the network parameters in order to minimize the sum-squared error:

$$F(x) = \sum_{q=1}^{Q} e_q^2 = \sum_{q=1}^{Q} (t_q - a_q)^2 \tag{3.10}$$

Where $\mathbf{x}$ is a vector containing all network weights and biases. If the network has multiple outputs this generalizes to

$$F(x) = \sum_{q=1}^{Q} e_q^T e_q = \sum_{q=1}^{Q} (t_q - a_q)^T (t_q - a_q) \tag{3.11}$$

Using a stochastic approximation, it will replace the sum-squared error by the inaccuracies on the latest target:

$$\hat{F}(x) = (t(k) - a(k))^T (t(k) - a(k)) = e^T(k)\, e(k) \tag{3.12}$$

where the expectation of the squared error has been replaced by the squared error at iteration k. The steepest descent algorithm for the approximate mean square error is:

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^{m}} \tag{3.13}$$

$$b_{i}^{m}(k+1) = b_{i}^{m}(k) - \alpha \frac{\partial \hat{F}}{\nabla b_{i}^{m}} \tag{3.14}$$

where a is the learning rate.

For a single-layer linear network, these partial derivatives in Equations (3.13) and (3.14) are conveniently computed, since the error can be written as an explicit linear function of the network weights. For the multilayer network, the error is not an explicit function of the weights in the hidden layers; therefore, these derivatives are not computed so easily. Because the error is an indirect function of the weights in the hidden layers, it will be use the chain rule of calculus to calculate the derivatives in Equations (3.13) and (3.14):

$$\frac{\partial \hat{F}}{\partial w_{i,j}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} \times \frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} \tag{3.15}$$

$$\frac{\partial \hat{F}}{\partial b_{i}^{m}} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} \times \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \tag{3.16}$$

The second term in each of these equations can be easily computed, since the net input to layer m is an explicit function of the weights and bias in that layer: The second term in each of these equations can be easily computed, since the net input to layer m is an explicit function of the weights and bias in that layer:

$$n_{i}^{m} = \sum_{j=1}^{s^{m-1}} w_{i,j}^{m} a_{j}^{m-1} + b_{i}^{m} \tag{3.17}$$

Therefore

$$\frac{\partial n_{i}^{m}}{\partial w_{i,j}^{m}} = a_{j}^{m-1}, \ \frac{\partial n_{i}^{m}}{\partial b_{i}^{m}} \tag{3.18}$$

If it is now define:

$$s_{i}^{m} = \frac{\partial \hat{F}}{\partial n_{i}^{m}} \tag{3.19}$$

(the sensitivity of ˆF to changes in the itˆh element of the net input at layer m), then Equations (3.15) and (3.16) can be simplified to:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^{m}} = s_i^m a_j^{m-1} \tag{3.20}$$

It can be now express the approximate steepest descent algorithm as:

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \alpha s_i^m a_j^{m-1} \tag{3.21}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \tag{3.22}$$

In matrix form, this becomes:

$$W^m(k+1) = W^m(k) - \alpha s^m \left(a^{m-1}\right)^T \tag{3.23}$$

$$b^m(k+1) = b^m(k) - \alpha s^m \tag{3.24}$$

where the individual elements of $\mathbf{s}m$ are given by Equation (3.19).

In some procedure it is unfortunate that the algorithm that can be usually refer to as back-propagation, given by Equations (3.23) and (3.24), is indeed simply a steepest lineage algorithm. There are dozens other optimization algorithms that can use the back-propagation procedure, in which derivatives are processed from the vitality rank of the network to the first. For example, conjugate heap and quasi-Newton algorithms [14,16] are generally more efficient than steepest descent algorithms, and yet they tins use the same back-propagation procedure to compute the necessary derivatives. The Levenberg Marquardt algorithm is very efficient for training small to medium-size networks [6].

3.8 Nonlinear Organization Discovery Using Neural Networks

In the adaptive control of unknown and non-linear systems, the schemes must firstly be identified, that is, considering a configuration for the intrigue and guess its parameters in which it relates the system's input and authority with minimum error.

With glance to the capabilities of neural networks in general approximation, the main thing here is to use them in identifying nonlinear systems.

In this sub-section, four models, which were introduced in [6] for the representation of a single-input single output (SISO) nonlinear plant, are presented. These models were chosen both for their generality as well as for analytical tractability. The models are motivated by corresponding models, which have been used in the adaptive systems literature for the identification of linear systems and can be considered as their generalizations to nonlinear systems. Since back-propagation is the principal method that we shall use for the adjustment of parameters of the identification model, the parameterization of the plant (and hence the model) is such as to make the application of the procedure relatively straightforward.

The replica of the four classes of plants introduced here can be described by the following nonlinear difference equations:

$$MODEL\,I:\ y_p(k+1) = \sum_{i=0}^{n-1} a_i y_p(k-i) + g\left[u(k), u(k-1), ..., u(k-m+1)\right] \tag{3.25}$$

$$MODEL\,II:\ y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p(k-n+1)\right] + \sum_{i=0}^{m-1} \beta u(k-i) \tag{3.26}$$

$$MODEL\,III:\ y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p(k-n+1)\right] + g\left[u(k), u(k-1), ..., u(k-m+1)\right] \tag{3.27}$$

$$MODEL\,IV:\ y_p(k+1) = f\left[y_p(k), y_p(k-1), ..., y_p(k-n+1), u(k), u(k-1), ..., u(k-m+1)\right] \tag{3.28}$$

Where [(k), (k)] represents the input-output pair of the SISO plant at time k. The functions f and g are assumed to be differentiable functions of their arguments. It is evident that Model IV subsumes Models I-III. However, Model IV is analytically the least tractable and hence for practical applications, some of the other models might prove more attractive. In this chapter, each of these models is briefly described.

Model I: The capability of the unknown nonlinear fortification in this argument is assumed to depend linearly on its past values and nonlinearly on the past values of the input. The latter is realized as shown in Fig. 3.13 and consists of tapped delay lines at the input and the critique path.

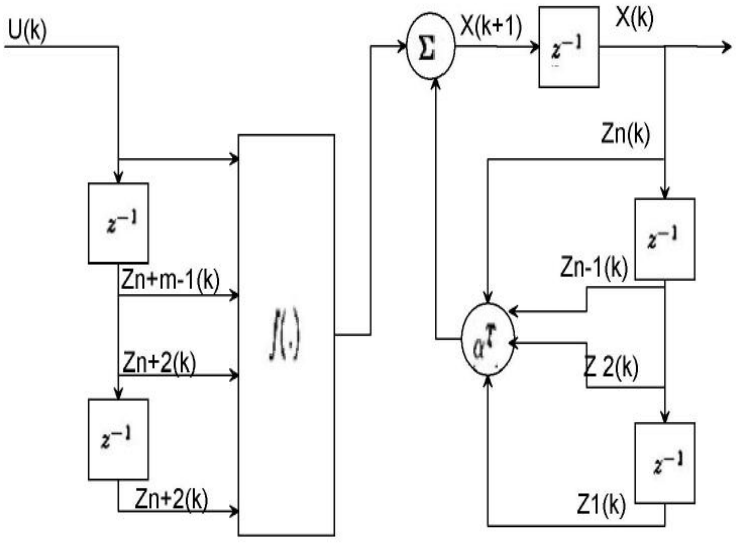

Figure 3.13: Structure of the plant for Model I [41]

Model II: This model is realized as shown in Fig. 3.14. In this case, the output depends linearly on the input (k) and its past values but nonlinearly on its own past values. The advantage of this model is that it lends itself readily to control in practical situation.

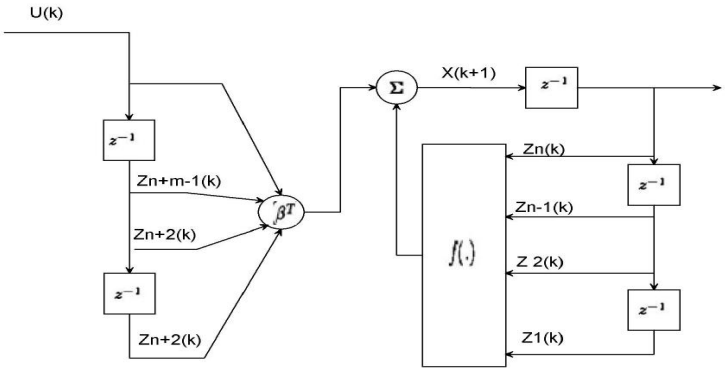

Figure 3.14: Structure of the plant for Model II [42]

Model III: The unknown plant in this box is described by a nonlinear unlikeness equation of the form:

$$x(k+1) = f[x(k), x(k-1), ..., x(k-n+1)] + g[u(k), u(k-1), ..., u(k-m+1)] \qquad (3.29)$$

And hence depends nonlinearly on both its past values as well as those of the input. However, the possessions of the power and output values are additive as shown in equation (3.29). The execution of equation (3.29) is shown in Fig. 3.15.

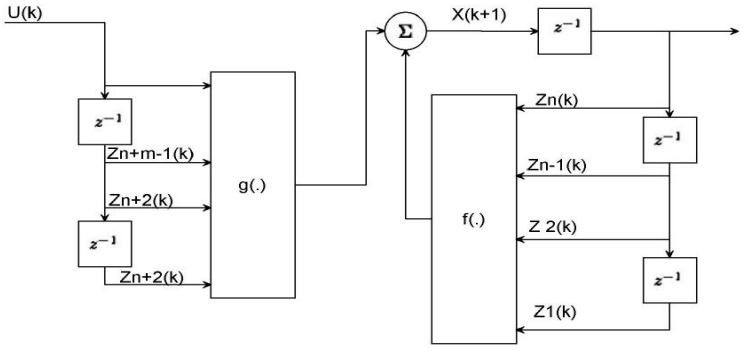

Figure 3.15: Structure of the plant for Model III [43]

Model IV: As mentioned earlier, this is the most general of all miniature introduced here and subsumes the earlier models. The ability at any instant in this matter is a

nonlinear role of the past values of both the capacity and the output. Once again, the execution of the shape using tapped suspension lines is shown in Fig. 3.16.

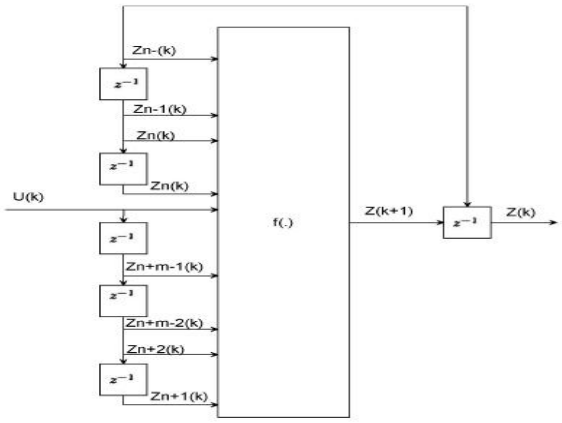

Figure 3.16: Structure of the plant for Model IV [44]

The identification ideal of the mill is composed of neural networks and tapped delay lines. In each case, the neural network is assumed to contain sufficient mathematics of layers, and complication in each layer, so as to be able to match exactly the input-output characteristics of the corresponding nonlinear mapping in the given plant. Firstly the grain of vista of mathematical analysis, this implies that the nonlinear functions in the difference equations describing the abuse can be replaced by neural networks with fixed but unknown weight mold. Hence, a theoretical breakdown to the adaptive discovery problem is assumed to exist from the outset.

To identify the plant, an identification model is chosen based on prior information concerning the class to which it belongs. For example, assuming that the plant has a structure described by Model III, the model is chosen to have the form shown in Fig. 3.12. The aim then is to determine the weights of the two neural networks $N1$ and $N2$. If $(k + 1)$ and $\hat{}x\ (k + 1)$ are respectively the outputs at stage $k + 1$ of the plant and the identification model, the error $e\ (k+1) = \hat{}x\ (k+1) — x\ (k+1)$ is used to update the weights of $N1$ and $N2$; static or dynamic back-propagation can

be used, depending on the structure of the identifier used.

1. Parallel Model: In this case, the structure of the identifier is identical to that of the plant with f and g replaced by $N2$ and $N1$ respectively. This is shown in Fig. 3.17. Since $N2$ is in a dynamic feedback loop, the parameters of $N1$ and N2 have to be adjusted using dynamic back-propagation.

2. Series-Parallel Model: In this case $(k+1)$ rather than $\hat{}x\ (k+1)$ is used to generate the output of the model. This implies that the model is described by the equation:

$$\hat{x}(k+1) = N_2[x(k), x(k-1), ..., x(k-n+1)] + N_1[u(k), u(k-1), ..., u(k-m+1)] \quad (3.30)$$

Since the model does not include a feedback flap containing a nonlinear element, static back-propagation rather than dynamic back-propagation of the error can be used to adjust the weights of the neural network.

The two methods outlined above have been discussed extensively in the context of the discovery of linear time-invariant orderliness with unknown parameters [5]. While the series- parallel agency has been shown to be globally stable, similar results are not available for the parallel model. To avoid many of the analytical hardship encountered, as well as to assure firmness and simplify the spotting procedure, the series-parallel model was used in [6]. Extensive computer simulations have revealed that a large class of nonlinear plants can be identified using the above procedure. However, theoretical studies concerning stability and convergence are still in the initial stages and numerous doubt have yet to be answered.

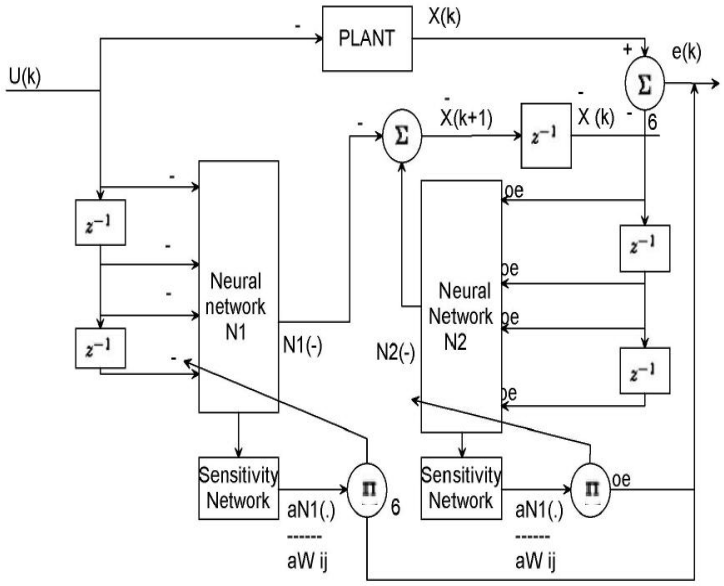

Figure 3.17: Structure of identification model for Model III [45]

3.9 NARMA Shape and its Approximations

The controller arrangement in general is dependent on the system's sighting model, so the system's spotting shape tins be considered so that a simple disposition supervisor tins be obtained.

The NARMA model is an exact representation of the nonlinear plant in a neighborhood of the equilibrium state. For reasons given toward the end of chapter 3, the model is not convenient for the computation of a control input to the plant to track a desired reference signal. In view of this, the proposed of two approximations to the NARMA model called the NARMA-L1 and the NARMA-L2 models. The main feature of these models is that the control input at time (the instant of interest in the control problem) occurs linearly in the equation relating inputs and outputs. This, in turn, permits easy algebraic computation of the control inputs without requiring a separate controller neural network. The fact that the use of neural networks is restricted to the identification model implies that only static gradient methods need

to be used. The equations for the two proposed approximate models are given below.

NARAM model:

$$y\left(k+d\right)=\overline{F}\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right),u\left(k\right),u\left(k-1\right),.....,u\left(k-n+1\right)\right] \tag{3.31}$$

Now, with Taylor expansion of $\overline{F}$ around the vector $\left(y\left(k\right),.,y\left(k-n+1\right),u\left(k\right),.,u\left(k-n+1\right)=0\right)$and then with the removal of high order sentences of the Taylor series expansion, the following models can be approximated, respectively:

NARAM_L1 model:

$$y\left(k+d\right)=f_o\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right)\right]+\sum_{i=0}^{n-1}g_i\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right)\right]u\left(k-i\right) \tag{3.32}$$

NARAM_L2 model:

$$y\left(k+d\right)=\overline{f_o}\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right)\right]+\overline{g_o}\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right),u\left(k-1\right),..,u\left(k-n+1\right)\right]u\left(k\right) \tag{3.33}$$

It is seen that f_0 and g_0 in the equation describing NARMA-L1 are only functions of the past values of the outputs, and $(k—1)\ldots$ $(k—n+1)$ as well as (k) occur linearly on the right-hand side (RHS) of (3.32). In contrast to this, NARMA-L2 model is described by only two terms in the RHS of (3.33) where both $\overline{f_o}$ and $\overline{g_o}$ are functions of $y(k)$,$y(k—1)$,$\ldots$,$y(k—n+1)$ and $u(k)$,$u(k—1)$,$\ldots$,$u(k—n+1)$ respectively.

CHAPTER 4

METHODOLOGY

4.1 Nerves System Based Arm Position Sensor System Description

Nerves system based arm position sensor is a device which senses the electrical pulse signal of the nerve and compares the desired arm position and responce arm position of the nerve defected arm. Nerves system based arm position sensor is a type of neuromodulation therapy in which electrodes are surfacely placed next to a selected peripheral nerve considered to be the source of nerve pain. One way of trying to control the arm position is that arises from peripheral nerves calls for a device that sends low levels of electricity to stimulate part(s) of the nerve. This electrical voltage is thought to interfere with how the nerve transmits the voltage impulse signals and responce through arm motion.

The mathimatical description of the system is shown bellow

$$\frac{d^2y\left(t\right)}{dt^2} = -\gamma + \frac{\alpha}{s}\frac{v^2\left(t\right)}{y\left(t\right)} - \frac{\beta}{s}\frac{dy\left(t\right)}{dt} \tag{4.1}$$

Where

$y\left(t\right)$	Arm postion output
$v\left(t\right)$	Impulse voltage input
γ	Arm postion acceleration
s	Device sensetivity function
α	Nerve transmisson coeficient
β	Nerve delay coeficient

The block diagram of the nerves system based arm position sensor device is shown in Figure 4.1 bellow.

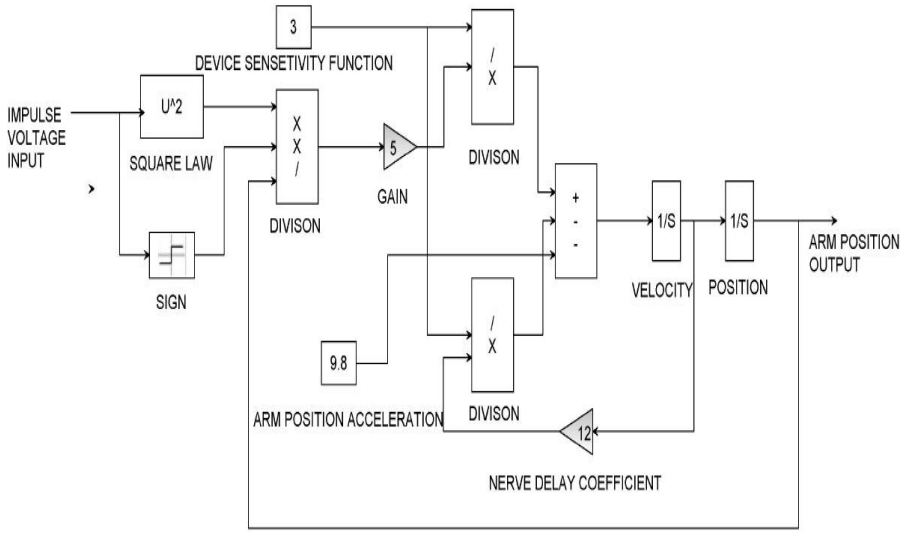

Figure 4.1: Block diagram of the nerves system based arm position sensor device

Figure 4.2: The nerves system based arm position sensor connection to human arm

4.2 Design of NARMA-L2 Neural Network Controller

The neurocontroller described in this section is referred to by two different names: response linearization control and NARMA-L2 control. It is referred to as feedback linearization when the plant shape has a particular form (companion form). It is referred to as NARMA-L2 control when the fortification mold can be approximated by the same form. The central theory of this type of control is to transform nonlinear design system into linear dynamics by canceling the nonlinearities. This section begins by submitting the companion system form and presentation how you can use a neural network to identify this model. Then it describes how the identified neural network model can be used to develop a controller.

4.2.1 Identification of the NARMA-L2 Model

The first step in using feedback linearization (or NARMA-L2) control is to identify the design to be controlled. You train a neural network to represent the forward dynamics of the system. The first step is to choose a patterns arrangement to use. One standard patterns that is used to represent general discrete-time nonlinear procedure is the nonlinear autoregressive-moving average (NARMA) model:

$$y\left(k+d\right)=N\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right),u\left(k\right),u\left(k-1\right),....,u\left(k-n+1\right)\right] \tag{4.2}$$

where u(k) is the system input, and y(k) is the system output. For the identification phase, you could train a neural network to approximate the nonlinear function N. If you want the system output to follow some reference trajectory $y\left(k+d\right)=y_r\left(k+d\right)$ the next step is to develop a nonlinear controller of the form:

$$u\left(k\right)=G\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right),y_r\left(k+d\right),u\left(k-1\right),....,u\left(k-m+1\right)\right] \tag{4.3}$$

The problem with using this controller is that if you want to train a neural network to create the function G to minimize mean square error, you need to use dynamic backpropagation. This can be quite slow. One solution is to use approximate models to represent the system. The controller used in this section is based on the NARMA-L2 approximate model:

$$\hat{y}\left(k+d\right)=f\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right),u\left(k-1\right),..,u\left(k-m+1\right)\right]$$

$$+g\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right),u\left(k-1\right),..,u\left(k-m+1\right)\right]u\left(k\right) \tag{4.4}$$

This model is in companion form, where the next controller input u(k) is not contained inside the nonlinearity. The advantage of this form is that you can solve for the control input that causes the system output to follow the reference $y\left(k+d\right)=y_r\left(k+d\right)$. The resulting controller would have the form

$$u\left(k\right)=\frac{\left[y_r\left(k+d\right)-f\left(y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right),u\left(k-1\right),....,u\left(k-n+1\right)\right)\right]}{g\left[y\left(k\right),y\left(k-1\right),...,y\left(k-n+1\right),u\left(k-1\right),....,u\left(k-n+1\right)\right]} \tag{4.5}$$

Using this equation directly can cause realization problems, because you must determine the control input u(k) based on the output at the same time, y(k). So, instead, use the model

$$y\left(k+d\right)=f\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right),u\left(k-1\right),..,u\left(k-m+1\right)\right]$$

$$+g\left[y\left(k\right),y\left(k-1\right),..,y\left(k-n+1\right),u\left(k-1\right),..,u\left(k-m+1\right)\right]u\left(k+1\right) \tag{4.6}$$

where d $\geq$ 2. Figure 4.3 shows the structure of a neural network representation

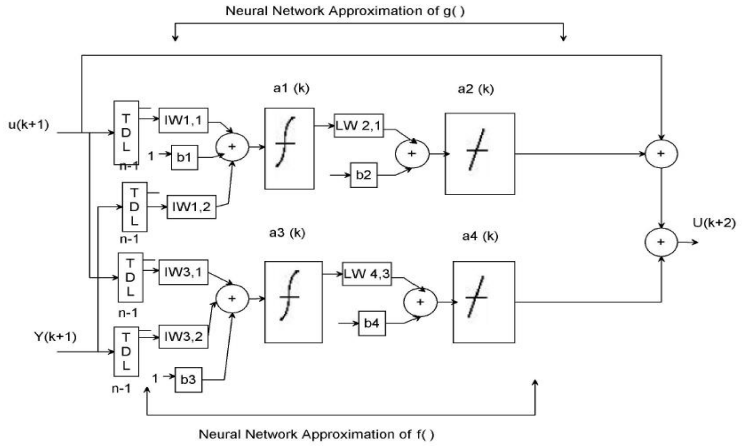

Figure 4.3: The structure of a neural network representation.

Using the NARMA-L2 model, you can obtain the controller

$$u(k+1) = \frac{[y_r(k+d) - f(y(k), y(k-1), ..., y(k-n+1), u(k), ..., u(k-n+1))]}{g[y(k), y(k-1), ..., y(k-n+1), u(k), ..., u(k-n+1)]} \tag{4.7}$$

which is realizable for d $\geq$ 2. Figure 4.4 shows the block diagram of the NARMA-L2 controller.

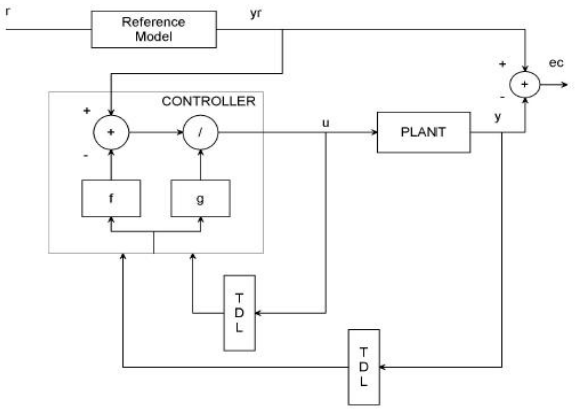

Figure 4.4: Block diagram of the NARMA-L2 controller

This controller can be implemented with the previously identified NARMA-L2 plant model, as shown in Figure 4.5 bellow

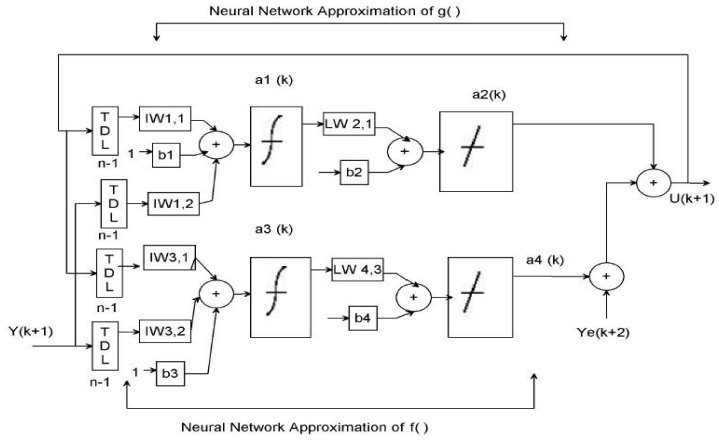

Figure 4.5: Previously identified NARMA-L2 plant model

The neural network arcticture, training data and training parameters is shown in the Table 4.1 bellow

Table 4.1: Neural network arcticture, training data and training parameters

Network Arcticture			
Size of hidden layer	9	No. delayed plant input	3
Sampling interval(sec)	0.01	No. delayed plant output	2
Training Data			
Training sample	10000	Maximum Plant output	Inf
Maximum Plant input	4	Minimum Plant output	0
Minimum Plant input	-1	Max interval value (sec)	1
Min interval value (sec)			0.1
Training Parameters			
Training Epochs			100

4.2.2 Levenberg-Marquardt Algorithm

Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has

the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as

$$H = JTJ$$

and the gradient can be computed as

$$g = JTe$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function is always reduced at each iteration of the algorithm. This algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights).

4.3 Design of NARMA-L2 Model Controller Using System Identification

The NARMA-L2 based neural network predictive controller uses a neural network model of a nonlinear plant to predict future plant performance. The controller then

calculates the control input that will optimize plant performance over a specified future time horizon. The first step in model predictive control is to determine the neural network plant model (system identification). Next, the plant model is used by the controller to predict future performance.

4.3.1 System Identification

The first stage of model NARMA-L2 based predictive control is to train a neural network to represent the forward dynamics of the plant. The prediction error between the plant output and the neural network output is used as the neural network training signal. The process is shown in Figure 4.6 bellow:

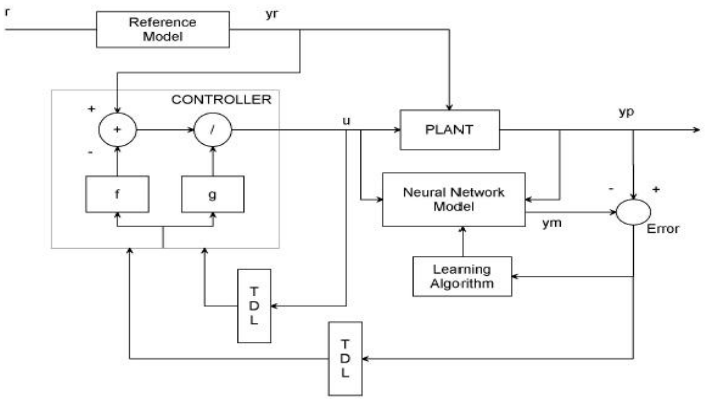

Figure 4.6: The process of training model predictive control

The neural network plant model uses previous inputs and previous plant outputs to predict future values of the plant output. The structure of the neural network plant model is given in Figure 4.7 bellow.

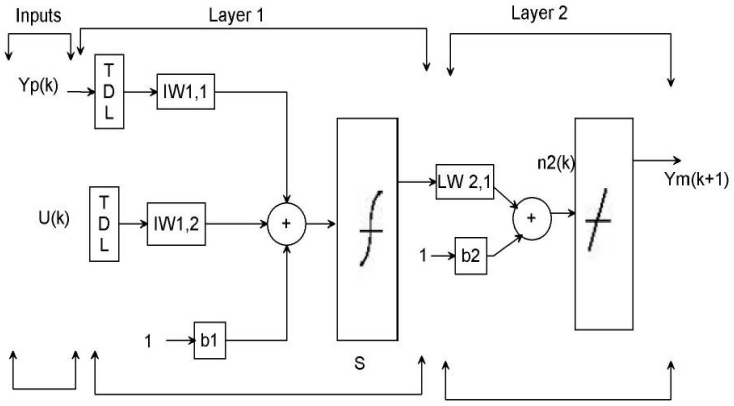

Figure 4.7: The structure of the neural network plant model

4.3.2 Predictive Control

The model predictive control method is based on the receding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization program to determine the control signal that minimizes the following performance criterion over the specified horizon

$$J = \sum_{j=N_1}^{N_2} (y_r(k+j) - y_m(k+j))^2 + \rho \sum_{j=1}^{N_u} (u'(k+j-1) - u'(k+j-2))^2 \qquad (4.8)$$

where N1, N2, and Nu define the horizons over which the tracking error and the control increments are evaluated. The u´ variable is the tentative control signal, yr is the desired response, and ym is the network model response. The ρ value determines the contribution that the sum of the squares of the control increments has on the performance index.

The following block diagram illustrates the model predictive control process. The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of u´ that minimize J, and then the optimal u is input to the plant.

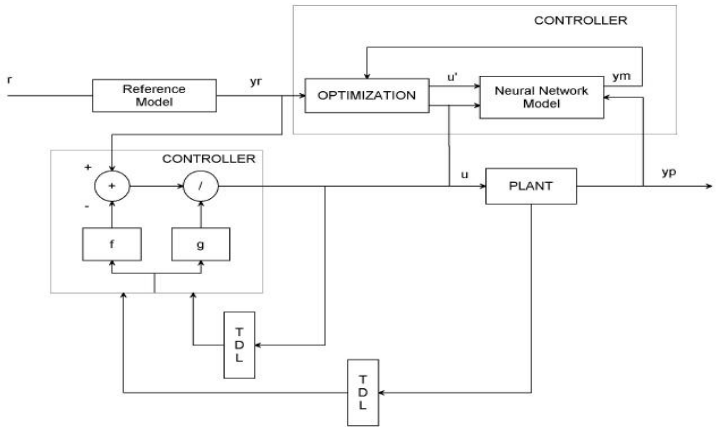

Figure 4.8: Block diagram of the model predictive control process

Table 4.2 shows the parameters for designing predictive controller

Table 4.2: Parameters for designing predictive controller

Neural Network Predictive Control			
Cost Horizon (N2)	7	Control Wieghting Function ρ	0.05
Control Horizon (Nu)	2	Search Parameter α	0.001
Iteration Per Sample Time			2

4.4 Design of NARMA-L2 Model Controller Using Adaptive Control

The NARMA-L2 based neural model reference adaptive control architecture uses two neural networks: a controller network and a plant model network, as shown in Figure 4.9 below. The plant model is identified first, and then the controller is trained so that the plant output follows the reference model output.

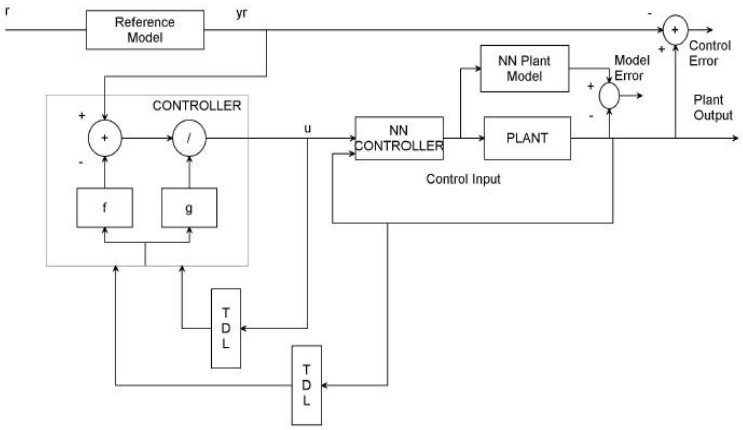

Figure 4.9: Block diagram of controller network and a plant model network

Figure 4.10 shows the details of the neural network plant model and the neural network controller. Each network has two layers, and you can select the number of neurons to use in the hidden layers. There are three sets of controller inputs:

- Delayed reference inputs
- Delayed controller outputs
- Delayed plant outputs

For each of these inputs, you can select the number of delayed values to use. Typically, the number of delays increases with the order of the plant. There are two sets of inputs to the neural network plant model:

- Delayed controller outputs
- Delayed plant outputs

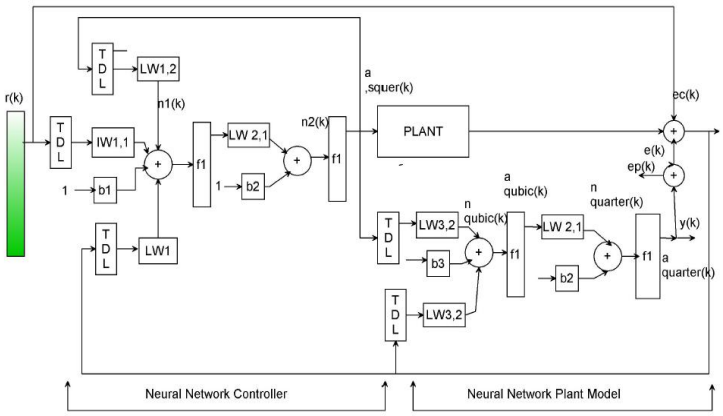

Figure 4.10: The neural network plant model and the neural network controller

CHAPTER 5

RESULTS AND DISSCUSION

This chapter basically focuses on the comparison of neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control . These three systems input is the desired arm position and they generate impulse voltage signal to be given to the arm and the arm deliver arm output position. These three systems tested with step, sine wave and random desired arm position signal.

5.1 Comparison of the Proposed Controllers using step input signal

The simulink model for the comparison of neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using step input signal is shown in the Figure 5.1 bellow

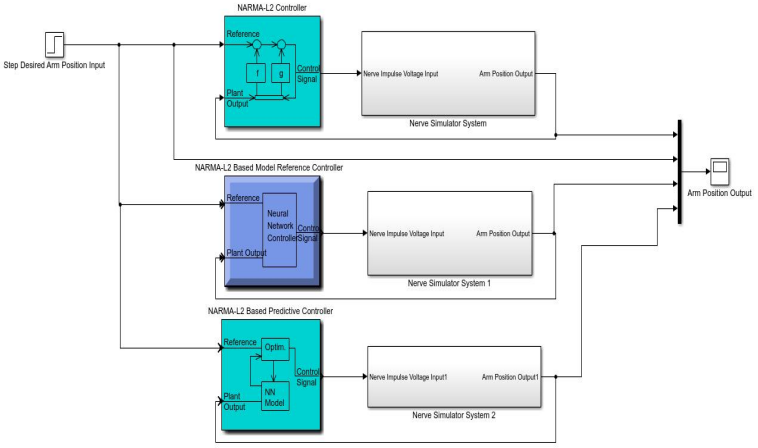

Figure 5.1: Step responce simulink model

The NARMA-L2 controller, NARMA-L2 based model refrence controller and NARMA-L2 based predictive controller subsystem is shown in Figure 5.2, Figure 5.3 and Figure 5.4 respectively.

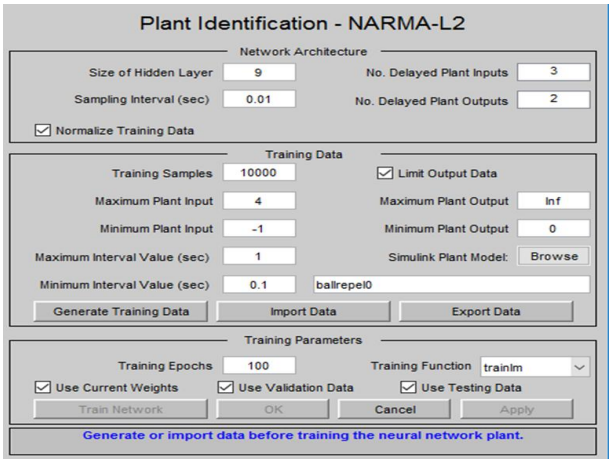

Figure 5.2: NARMA-L2 controller

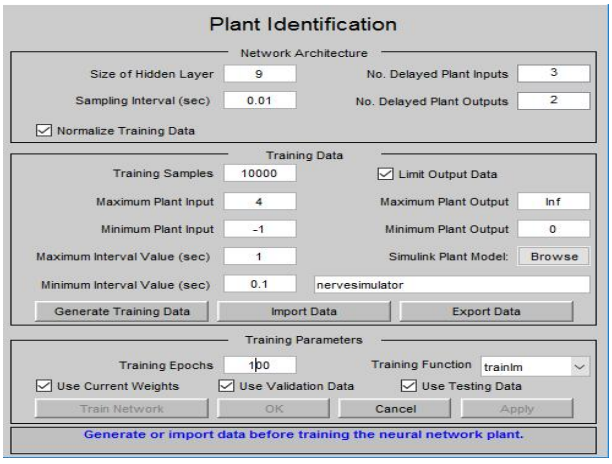

Figure 5.3: NARMA-L2 based model refrence controller

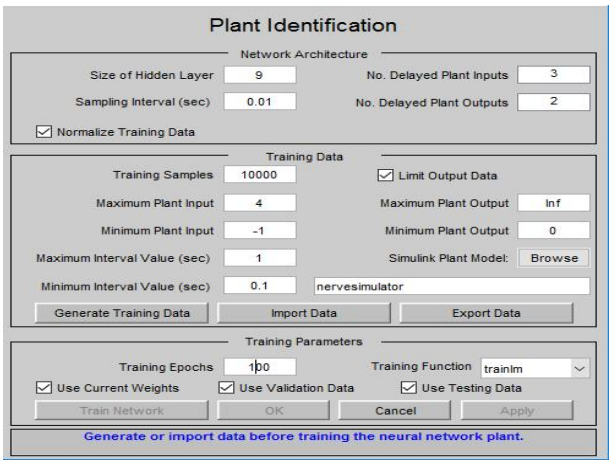

Figure 5.4: NARMA-L2 based predictive controller

The simulation output for the comparison of the neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using step input signal is shown in the Figure 5.5 bellow.

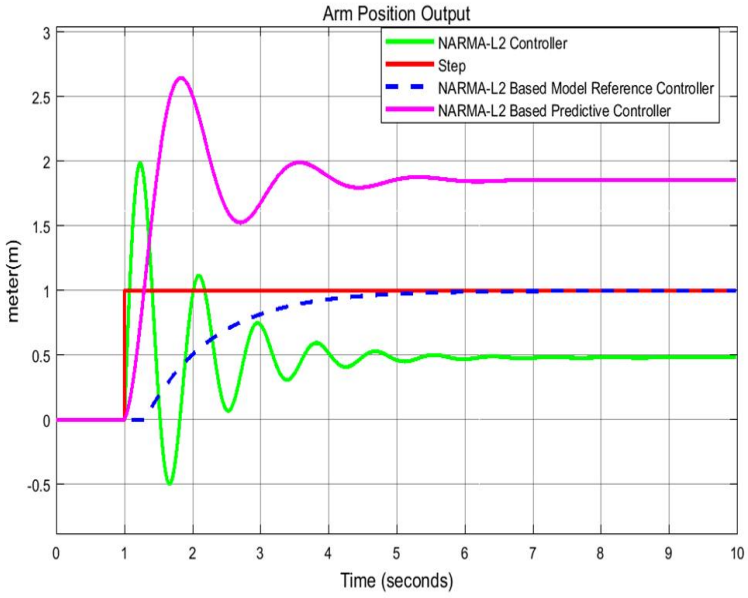

Figure 5.5: Step responce simulation result

The neural network controller with NARMA-L2 controller has a high percentage overshoot and since the step input signal is the desired arm position of the patient and the output of this controller has a steady state value of 0.5 m while the desired arm position is 1m. This show us that the neural network controller with NARMA-L2 controller doesnot feed enogh nerve impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model system identification based predictive controller has a bigger percentage overshoot and since the step input signal is the desired arm position of the patient and the output of this controller has a steady state value of 1.75 m while the desired arm position is 1m. This show us that the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model based model reference adaptive control has a no percentage overshoot and since the step input signal is the desired arm position of the patient and the output of this controller has the same steady state value of 1 m with the desired arm position. This show us that the neural network controller with NARMA-L2 model based model reference adaptive control gives the exact nerve impulse voltage to the nerve of the arm.

5.2 Comparison of the Proposed Controllers using Sine Wave Input Signal

The simulink model for the comparison of neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using sine wave input signal is shown in the Figure 5.6 bellow

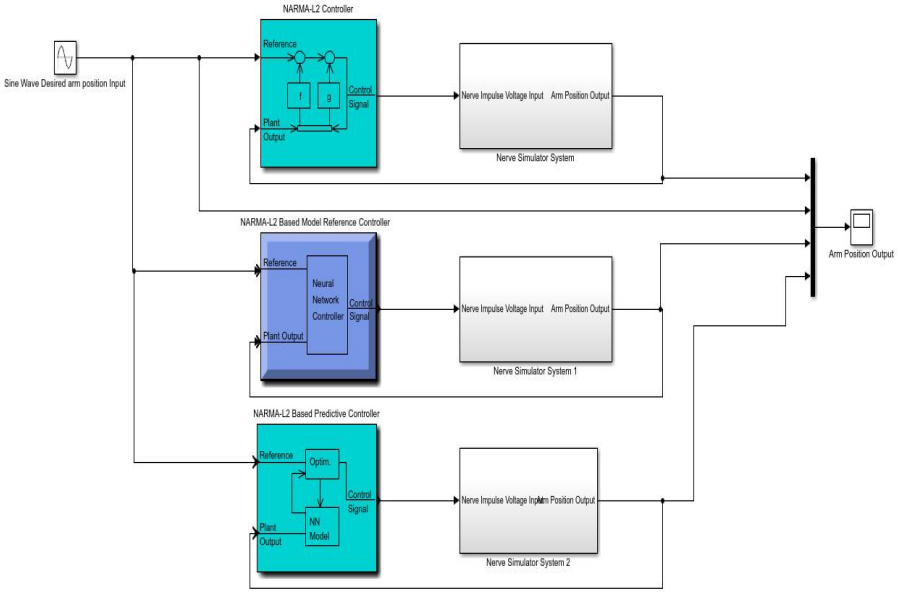

Figure 5.6: Sine wave responce simulink model

The simulation output for the comparison of the neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using sine wave input signal is shown in the Figure 5.7 bellow

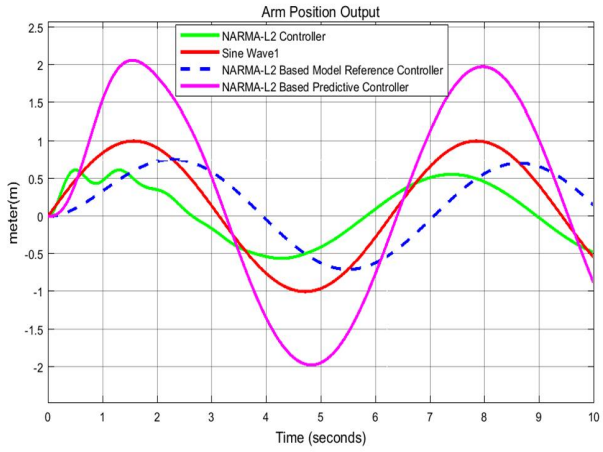

Figure 5.7: Sine wave responce simulation result

The neural network controller with NARMA-L2 controller sine wave input signal is the desired arm position of the patient and the output of this controller has a peak value of 0.6 m while the desired arm position is 1m. This show us that the neural network controller with NARMA-L2 controller gives low impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model system identification based predictive controller sine wave input signal is the desired arm position of the patient and the output of this controller has a peak value of 2.1 m while the desired arm position is 1m. This show us that the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model based model reference adaptive control sine wave input signal is the desired arm position of the patient and the output of this controller has a peak value of 0.8 m while the desired arm position is 1m. This show us that the neural network controller with NARMA-L2 model based model reference adaptive control gives almost the exact nerve impulse voltage to the nerve of the arm.

5.3 Comparison of the Proposed Controllers using Random Input Signal

The simulink model for the comparison of neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using random input signal is shown in the Figure 5.8 bellow

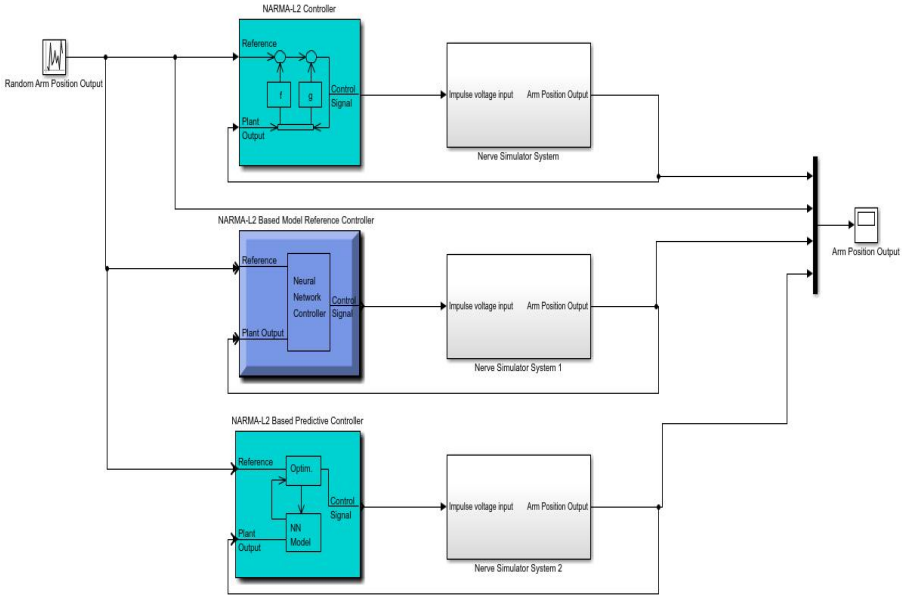

Figure 5.8: Random input responce simulink model

The simulation output for the comparison of the neural network controller with NARMA-L2 model, neural network controller with NARMA-L2 model system identification based predictive controller and neural network controller with NARMA-L2 model based model reference adaptive control using random input signal is shown in the Figure 5.9 bellow

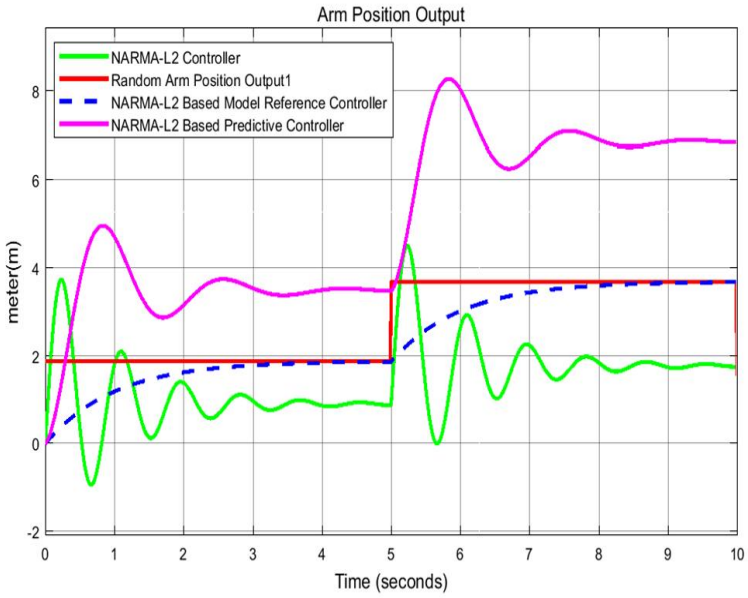

Figure 5.9: Random input responce simulation result

The neural network controller with NARMA-L2 controller random input signal is the desired arm position of the patient and the output of this controller has a peak value of 4.4 m while the desired arm position is 3.5m. This show us that the neural network controller with NARMA-L2 controller gives high impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model system identification based predictive controller random input signal is the desired arm position of the patient and the output of this controller has a peak value of 8.5 m while the desired arm position is 3.5 m. This show us that the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm. The neural network controller with NARMA-L2 model based model reference adaptive control random input signal is the desired arm position of the patient and the output of this controller has a peak value of 3.5 m while the desired arm position is 3.5 m. This show us that the neural network controller with NARMA-L2 model based model reference adaptive control gives the exact nerve impulse voltage to the nerve of the arm.

5.4 Numerical Value Comparison of the Proposed systems

5.4.1 Step Input

The numerical value comparison of the proposed systems for step input is shown in the Table 5.1 bellow.

Table 5.1: Numerical value comparison of the proposed systems for step Input

No	System	NARMA-L2 model	Model reference	Predictive controller
1	Peak Overshot	40%	0%	55%
2	Settling Time	6	4.3	5.5
3	Steady state value	0.5	1	1.75

The result from Table 5.1 shows us that the neural network controller with NARMA-L2 model based model reference adaptive control show the best responce.

CHAPTER 6

CONCLUSIONS & RECOMMENDATION

6.1 Conclusions

In this thesis, nerves system based arm position sensor device is used to measure the exact arm position for nerve patients using the proposed systems.Three diffrent systems are proposed which are a neural network controller is designed with NARMA-L2 model, neural network controller is designed with NARMA-L2 model system identification based predictive controller and neural network controller is designed with NARMA-L2 model based model reference adaptive control system.

The proposed controllers are tested for comparing the actual and desired arm position of the nerves system based arm position sensor device for three desired arm position inputs (step, sine wave and random).

The simulation result for a step input signals shows that the neural network controller with NARMA-L2 controller feeds a high impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model based model reference adaptive control gives the exact nerve impulse voltage to the nerve of the arm.

The simulation result for a sine wave input signals shows that the neural network controller with NARMA-L2 controller gives high nerve impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model based model reference adaptive control gives the exact nerve impulse voltage to the nerve of the arm.

The simulation result for a random input signals shows that the neural network controller with NARMA-L2 controller gives high nerve impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model system identification based predictive controller feed excess nerve impulse voltage to the nerve of the arm and the neural network controller with NARMA-L2 model based model reference adaptive control gives the exact nerve impulse voltage to the nerve of the arm.

Finally, the comparative simulation result prove the effectivness of the presented neural network controller with NARMA-L2 model based model reference adaptive control and it achieves to balance between the actual and desired arm position tests for the nerve patient by adjusting the nerve impulse voltage given to the arm.

6.2 Recommendation

The future work presented as follows

1. In this thesis we use NARMA-L2 model. The next step is to use NARMA-L3 model and NARMA-L4 model.

2. In this thesis, the nerves system based arm position sensor device is designed to give a one dimensional output. so the next step is to design a device with a three dimensional output.

3. The main advantage of the computer simulation is convenient to operate, but its disadvantage is also very obvious, which the simulation results are not withstand the actual test. The next step is to utility experiment scheme to experiment and verify the simulation results. This process can further improve the simulation methods.

4. In this thesis we only established arm based nerves system sensor device. so the next step is to design a device for spinal cord nerves and even for the whole body parts.

5. NARMA-L2 control is used in the first step in using feedback linearization is to identify the system to be controlled. The next step is to train a neural network to represent the forward dynamics of the system.

REFERENCES

[1] T. A. Al-Zohary "Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks and Approximate Models" Computer and Systems Eng. Dept. Fac. Of Eng. Ain Shams University, 2018.

[2] Yousif Al-Dunainawi "A New MIMO ANFIS-PSO Based NARMA-L2 Controller for Nonlinear Dynamic Systems" IEEE Control Syst. Soc., vol. 1, p. 246, 2017.

[3] Cidambaram Vijay Nagaraj "Design of Buck Boost Converter for Nonlinear Systems using Adaptive Controller" Engineering and Technology, Pudukkottai-622507, Tamil Nadu, India, 2017.

[4] Mehdi Ramezani "Design of NARMA L-2 Control of Nonlinear Inverted Pendulum" International Research Journal of Applied and Basic Sciences, 2016

[5] NgocKhoat Nguyen "An Investigation of Intelligent Controllers Based on Fuzzy Logic and Artificial Neural Network for Power System Frequency Maintenance" Turkish Journal of Electrical Engineering & Computer Sciences, (2016) 24: 2893 – 2909.

[6] Dhanraj Suman "Implementation of Narma-L2 Controller for Magnetic Levitation System" 6th international conference on science technology and management, 04 Dec, 2016.

[7] Pratik Ghutke "Performance Analysis of Neural Network Based Narma Control for CSTR" International Journal for Innovative Research in Science & Technology— Volume 1 — Issue 8 — January 2015.

[8] Cabrera, J. B. D. and Narendra, K. S., On Regulation and Tracking in Nonlinear Discrete" Univ., New Haven, CT, Tech. Rep, 2015.

[9] Priyanka Sharma "NARMA-L2 Controller for Five-Area Load Frequency Control" Int. Jour. of App. Sc. and Eng. (IJASE) 2(2) : Dec 2014, 91-101.

[10] Razika Zamoum Boushaki "Artificial Neural Network Control of the Recycle Compression System" Studies in Informatics and Control, Vol. 23, No. 1, March 2014.

[11] Aminreza Riahi "2 Link Robot Controller Using A Fusion of Neural Network and PID Controller Design" International Journal of Artificial Intelligence and Mechatronic s Volume 2, Issue 3, ISSN 2320 – 5121, 2014.

[12] Priyanka Sharma "NARMA-L2 Controller for Five-Area Load Frequency Control" Proceedings of the IEEE 2014; 121(7):601–8.

[13] Ahmed M. Kassem "Nonlinear Autoregressive Moving Average Controller for Isolated Wind Generation System Control" Journal of Engineering Sciences, Assiut University, Vol. 40, No 1, pp.209-222, January 2012.

[14] Ahmed. M. Kassem "Modeling, Analysis and Neural MPPT Control Design of a PV Generator Powered DC Motor-Pump System" WSEAS TRANSACTIONS on SYSTEMS Issue 12, Volume 10, December 2011.

[15] Mahdi Vaezi "Adaptive Control of a Robotic Arm Using Neural Networks Based Approach" International Journal of Robotics and Automation, (IJRA), Volume: (1), Issue: (5), 2011.

[16] Yang, C "Output Feedback Adaptive Control of a Class of Nonlinear Discrete-Time Systems with Unknown Control Directions" Automatica 45 (2009) 270-276.

[17] Leila Fallah Araghi "Neural Network Controller for Two links- Robotic Manipulator Control with Different Load" Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol II IMECS 2009, March 18 - 20, 2009, Hong Kong.

[18] S.S. Mokri "Real Time Implementation of NARMA-L2 Control of a Single Link Manipulator" American Journal of Applied Sciences 5 (12): 1642-1649, 2008 ISSN 1546-9239 © 2008 Science Publications.

[19] Zhongsheng Hou "Model-Free Adaptive Control for a Class of Nonlinear Discrete-Time Systems Based on the Partial Form Linearization" Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

[20] Martin T. Hagan "An Introduction to the Use of Neural Networks in Control Systems" International Journal of Robust and Nonlinear Control Int. J. Robust Nonlinear Control 2002; 12:959–985.

[21] Noriega, J. R. and Wang, H., A Direct Adaptive Neural-Network Control for Unknown Nonlinear Systems and its Application, IEEE Trans. Neural Networks, vol. 9, pp. 27-33, Jan. 1998.

[22] Narendra, K. S. and Mukhopadhyay S Adaptive Control Using Neural Networks and Approximate Models, IEEE Trans. On Neural Networks, vol. 8, pp. 475-485, 1997.

[23] Levin, A. U. and Narendra, K. S., Control of Nonlinear Dynamical Systems Using Neural Networks: Part II Identification and Part III Control, IEEE Trans. Neural Networks, vol. 4, Mar. 1993, also IEEE Trans. Neural Networks, vol. 7, Jan. 1996, pp. 30-42; also Center Syst. Sci., Yale Univ., New Haven, CT, Tech. Rep. 9116 and 9117.

[24] Narendra, K. S. and Parthasarathy, K., Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks, IEEE Trans. Neural Networks, vol. 2, pp. 252-262, Mar. 1991.

[25] Narendra, K. S. and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans. Neural Networks, vol. 1, pp. 4-27, 1990.

[26] Jordan, M. I. and Jacobs, A., "Learning to Control an Unstable System with Forward Modeling," Advances in Neural Inform. Processing Syst., vol. 2, pp. 324-331, 1990.

[27] Saerens, M. and Soquet, A., "A Neural Controller," in Proc. 1st Int. Conf. Artificial Neural Networks, London, Oct. 1989, pp. 211-215.

[28] Narendra, K. S. and Annaswamy, A. M. "Stable adaptive systems". Englewood Cliffs, NJ: Prentice-Hall.(1989).

[29] Figure 3.1. An information adapting unit, visualized as a graph. "Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[30] Figure 3.2. Linear regression. "Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[31] Figure 3.3. Logistic regression. "Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[32] Figure 3.4. Simplied anatomy of a logistic regression process. “Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[33] Figure 3.5. Simplied anatomy of a multi-class position network. “Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[34] Figure 3.6. A simple neural network with one hidden layer. “Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[35] Figure 3.7. The parameter matrix and the activation of the rst (or input) layer a in a neural network element. “Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[36] Figure 3.8. Example of the annotation used to number the complication of a neural network. “Machine Learning Mark K Cowan, The Creative Commons Attribution-NonCommercial-NoDerivs 3.0, 2013.

[37] Figure 3.9. (a) Single-input neuron, and (b) Log-sigmoid Transfer Function. “Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[38] Figure 3.10. (a) Multiple-input neuron, (b) Neuron with R inputs, form notation.. “Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[39] Figure 3.11. (a) Layer of S neurons, and (b) mold notation.. “Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[40] Figure 3.12. Three-layer network. “Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[41] Figure 3.13. Structure of the plant for Model I. “Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[42] Figure 3.14. Structure of the plant for Model II. "Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[43] Figure 3.15. Structure of the plant for Model III. "Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[44] Figure 3.16. Structure of the plant for Model IV. "Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

[45] Figure 3.17. Structure of identication model for Model III. "Adaptive Control Using NARMA-L2 Model for Nonlinear Systems Jamal Saeedi, Dalle Molle Institute for Artificial Intelligence, 2007.

A.1 Appendix Neural network Data

A.2 NARMA-L2 Neural Network Data

The neural network training data is shown bellow

	Input	Output	flag	sampling Time
1	-0.53801	-0.53801	-0.53801	0.01
2	-0.53801	-0.53801	-0.53801	
3	-0.53801	-0.53801	-0.53801	
4	-0.53801	-0.53801	-0.53801	
5	-0.53801	-0.53801	-0.53801	
6	-0.53801	-0.53801	-0.53801	
7	-0.53801	-0.53801	-0.53801	
8	-0.53801	-0.53801	-0.53801	
9	-0.53801	-0.53801	-0.53801	
10	-0.53801	-0.53801	-0.53801	
11	-0.53801	-0.53801	-0.53801	
12	-0.53801	-0.53801	-0.53801	
13	-0.53801	-0.53801	-0.53801	
14	-0.53801	-0.53801	-0.53801	
15	-0.53801	-0.53801	-0.53801	
16	-0.53801	-0.53801	-0.53801	
17	-0.53801	-0.53801	-0.53801	
18	-0.53801	-0.53801	-0.53801	
19	-0.53801	-0.53801	-0.53801	
20	-0.53801	-0.53801	-0.53801	
21	-0.53801	-0.53801	-0.53801	
22	-0.53801	-0.53801	-0.53801	
23	-0.53801	-0.53801	-0.53801	
24	-0.53801	-0.53801	-0.53801	
25	-0.53801	-0.53801	-0.53801	
26	-0.53801	-0.53801	-0.53801	
27	-0.53801	-0.53801	-0.53801	
28	-0.53801	-0.53801	-0.53801	
29	-0.53801	-0.53801	-0.53801	
30	-0.53801	-0.53801	-0.53801	
31	-0.53801	-0.53801	-0.53801	
32	-0.53801	-0.53801	-0.53801	
33	0.21824	0.21824	0.21824	
34	0.21824	0.21824	0.21824	
35	0.21824	0.21824	0.21824	
36	0.21824	0.21824	0.21824	
37	0.21824	0.21824	0.21824	
38	0.21824	0.21824	0.21824	
39	0.21824	0.21824	0.21824	
40	0.21824	0.21824	0.21824	
41	0.21824	0.21824	0.21824	
42	0.21824	0.21824	0.21824	
43	0.21824	0.21824	0.21824	
44	0.21824	0.21824	0.21824	

The testing, training and validation data is shown in Figure A.1, Figure A.2 and Figure A.3 respectively.

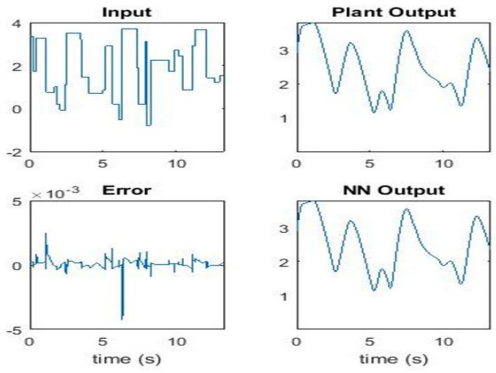

Figure A.1: Testing data for NN NARMA-L2

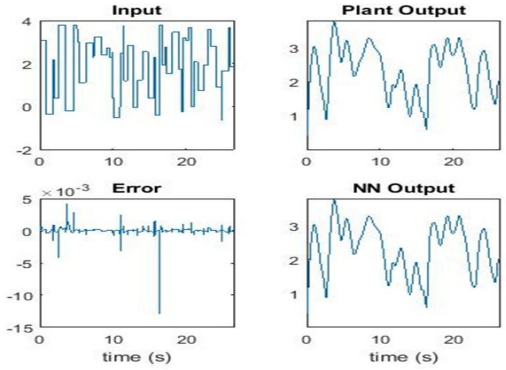

Figure A.2: Training data for NN NARMA-L2

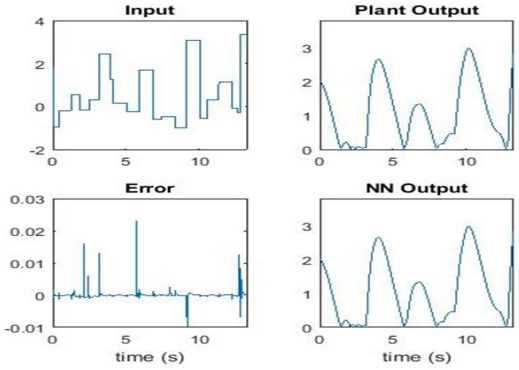

Figure A.3: Validation data for NN NARMA-L2

The neural network training algorism is shown in Figure A.4 bellow

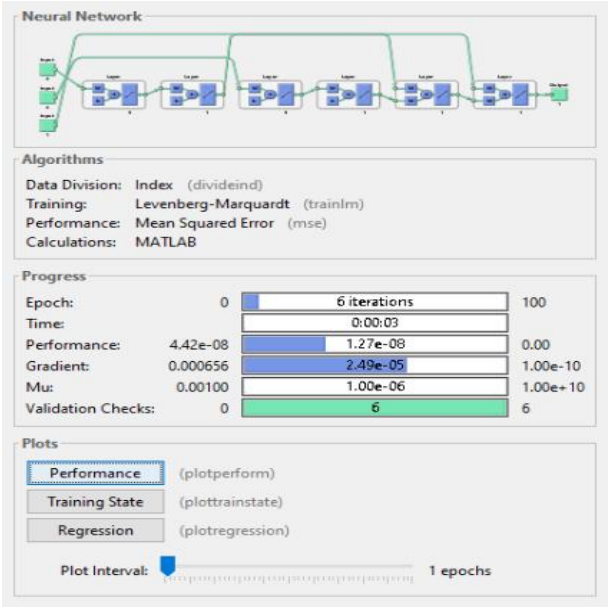

Figure A.4: Neural network training

The neural network training regression, training state and best validation performance is shown in Figure A.5, Figure A.6 and Figure A.7 respectivley.

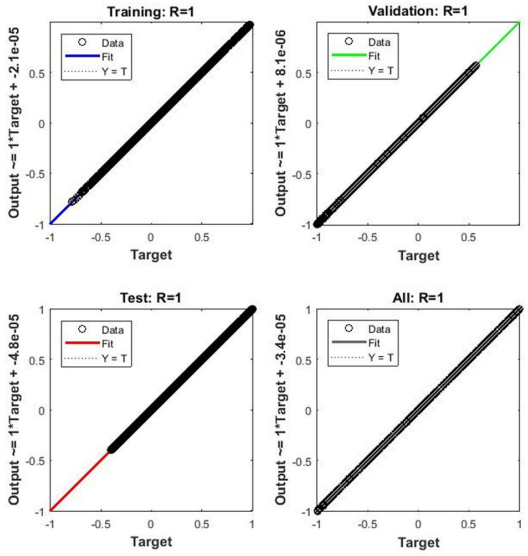

Figure A.5: The neural network training regression

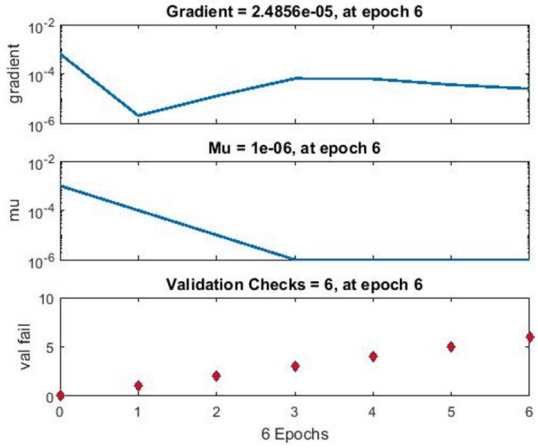

Figure A.6: The neural network training state

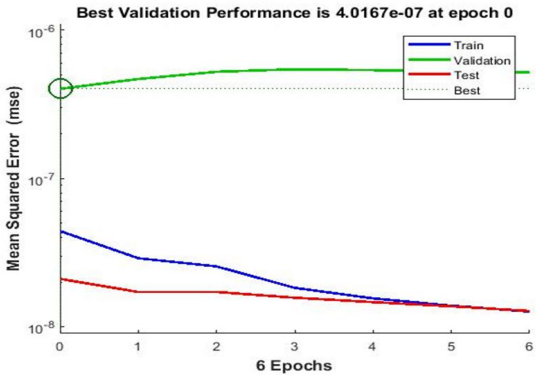

Figure A.7: The neural network best validation performance

A.3 NARMA-L2 Based System Identification Neural Network Data

The neural network training data is shown bellow

No	Input	output	sampling time
1	0.236845	0	0.05
2	0.236845	0.002412	
3	0.236845	0.008748	
4	0.236845	0.017867	
5	0.236845	0.028872	
6	0.236845	0.041059	
7	0.236845	0.053887	
8	0.236845	0.066939	
9	0.236845	0.079905	
10	0.236845	0.092556	
11	-0.18352	0.104727	
12	-0.18352	0.112025	
13	-0.18352	0.111698	
14	-0.18352	0.105728	
15	-0.18352	0.095693	
16	-0.18352	0.082841	
17	-0.18352	0.068153	
18	-0.18352	0.052388	
19	-0.18352	0.036133	
20	-0.18352	0.019828	
21	-0.18352	0.003805	
22	-0.18352	-0.0117	
23	-0.18352	-0.02652	
24	-0.18352	-0.04055	
25	-0.18352	-0.05371	
26	-0.18352	-0.06598	
27	-0.18352	-0.07735	
28	-0.18352	-0.08784	
29	-0.18352	-0.09746	
30	-0.18352	-0.10626	
31	0.674293	-0.11427	
32	0.674293	-0.11282	
33	0.674293	-0.09647	
34	0.674293	-0.0694	
35	0.674293	-0.03492	
36	0.674293	0.004382	
37	0.674293	0.046498	
38	0.674293	0.089878	
39	0.497732	0.133354	
40	0.497732	0.174259	
41	0.497732	0.210838	
42	0.497732	0.243503	
43	0.497732	0.272634	
44	0.497732	0.298582	

The testing, training and validation data is shown in Figure A.8, Figure A.9 and Figure A.10 respectively.

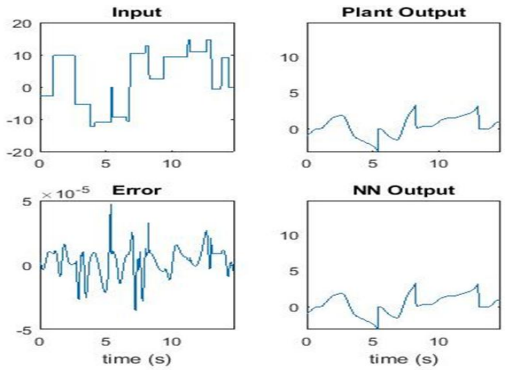

Figure A.8: Testing data for NN model refrence control

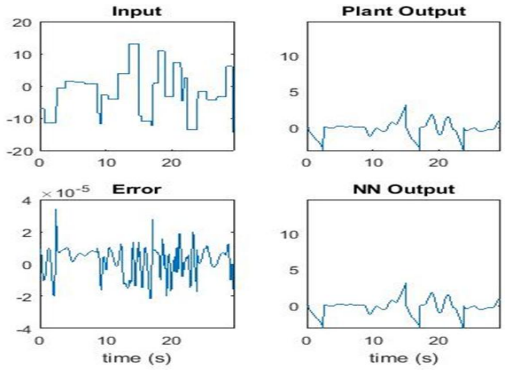

Figure A.9: Training data for NN model refrence control

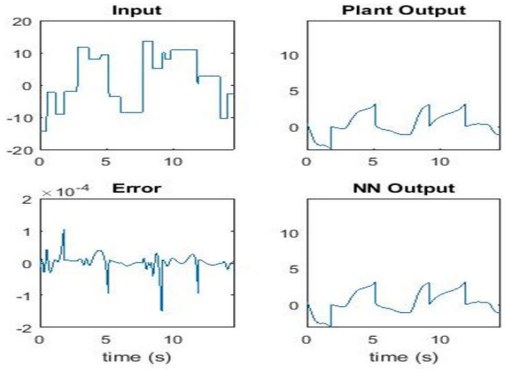

Figure A.10: Validation data for NN model refrence control

The neural network training algorism is shown in Figure A.11 bellow

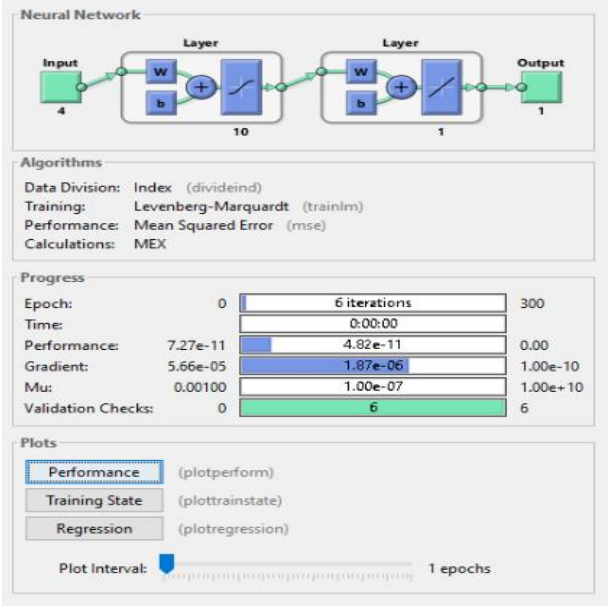

Figure A.11: Neural network training

The neural network training regression, training state and best validation performance is shown in Figure A.12, Figure A.13 and Figure A.14 respectivley.

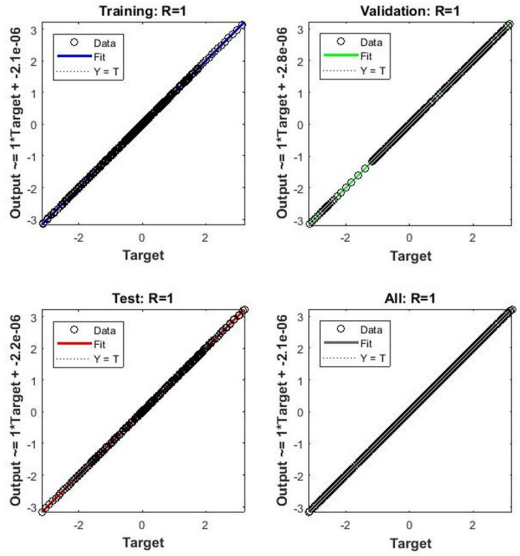

Figure A.12: Neural network training regression

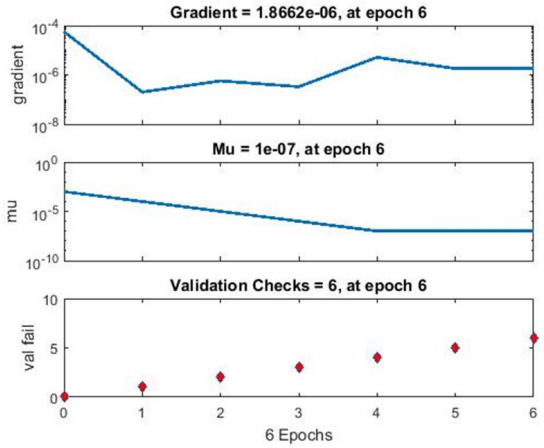

Figure A.13: Neural network training state

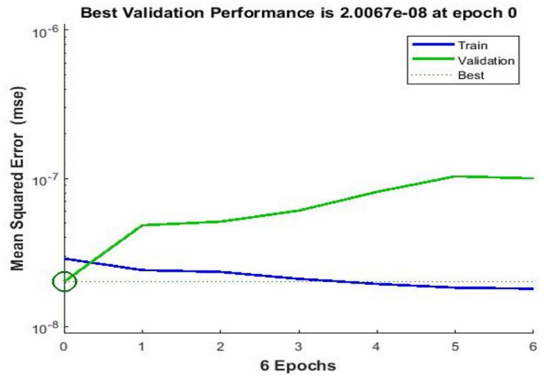

Figure A.14: Neural network training performance

A.4 NARMA-L2 Based Adaptive Control Neural Network Data

The neural network training data is shown bellow

No	Input	output	flag	sampling Time
1	1.303335	22	1	0.2
2	1.303335	22.00226	1	
3	1.303335	22.00448	1	
4	1.303335	22.00665	1	
5	1.303335	22.00879	1	
6	1.303335	22.01089	1	
7	1.303335	22.01294	1	
8	1.303335	22.01496	1	
9	1.303335	22.01694	1	
10	1.303335	22.01888	1	
11	1.303335	22.02079	1	
12	1.303335	22.02266	1	
13	1.303335	22.02449	1	
14	1.303335	22.02629	1	
15	1.303335	22.02805	1	
16	1.303335	22.02978	1	
17	1.303335	22.03148	1	
18	1.303335	22.03314	1	
19	1.303335	22.03477	1	
20	1.303335	22.03636	1	
21	1.303335	22.03793	1	
22	1.303335	22.03946	1	
23	1.303335	22.04096	1	
24	1.303335	22.04243	1	
25	1.303335	22.04387	1	
26	1.303335	22.04528	1	
27	1.303335	22.04667	1	
28	1.303335	22.04802	1	
29	1.303335	22.04934	1	
30	1.303335	22.05064	1	
31	1.303335	22.05191	1	
32	1.303335	22.05315	1	
33	1.303335	22.05436	1	
34	1.303335	22.05555	1	
35	1.303335	22.05671	1	
36	1.303335	22.05785	1	
37	1.303335	22.05896	1	
38	1.303335	22.06004	1	
39	1.303335	22.06111	1	
40	1.303335	22.06214	1	
41	1.303335	22.06315	1	
42	1.303335	22.06414	1	
43	1.303335	22.06511	1	
44	1.303335	22.06605	1	

The testing and validation data is shown in Figure A.15 and Figure A.16 respectively.

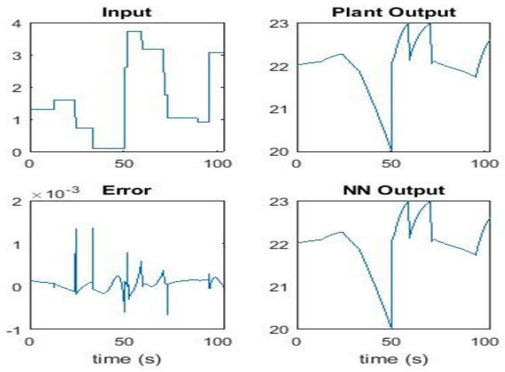

Figure A.15: Traing data for NN predictive control

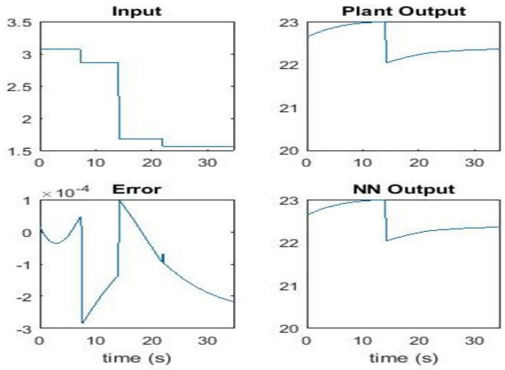

Figure A.16: Validation data for NN predictive control

The neural network training algorism is shown in Figure A.17 bellow

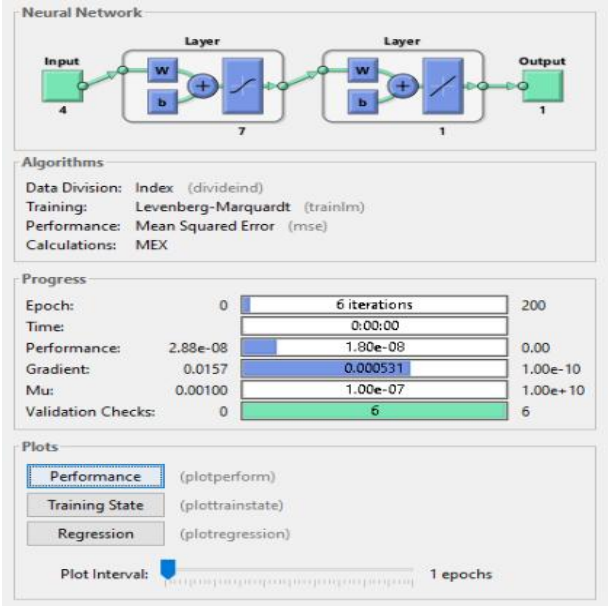

Figure A.17: Neural network training algorism

The neural network training regression, training state and best validation performance is shown in Figure A.18, Figure A.19 and Figure A.20 respectivley.

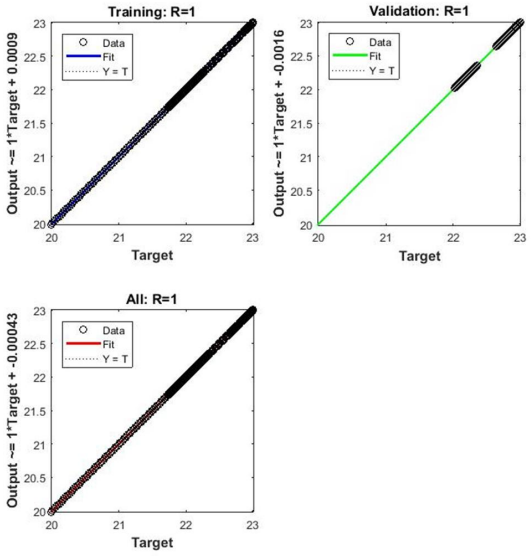

Figure A.18: Neural network teaing regression

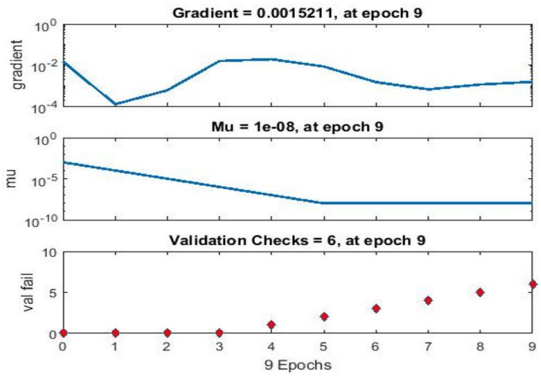

Figure A.19: Neural network traing state

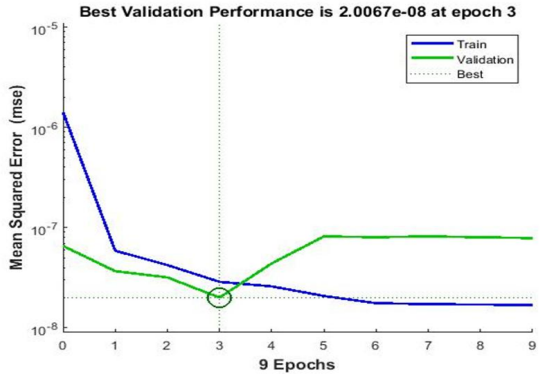

Figure A.20: Neural network traing performance