Study on application of Machine Learning Intelligence for Nonlinear Dynamical Systems

Abstract:

In the literature, machine learning has been referred to as deeply structured learning, hierarchical learning and feature based learning which can provide computational models from high-level data abstractions. One of the most used learning structures is the multiple-layered models of inputs, commonly known as neural networks, which comprise multiple levels of non-linear operations. The machine learning algorithms are able solve many problems around fault detection, isolation and recovery.

There has been a growing interest in using learning architectures in advanced robotics applications, e.g., object detection, scene semantic segmentation, and grasping. The real-time learning of high-dimensional features for robotics applications via machine learning techniques is another important topic. In addition, other topics in robotics such as obstacle detection and context-dependent social mapping are also being addressed by researchers through machine learning methods.

Machine learning algorithms provide real time driving decisions for automated vehicles (self-driving vehicles or driverless cars) from integration of numerous sensors onboard the vehicle. The advancement of autonomous navigation and situational awareness systems adapt neural networks for analyzing the multi-modal sensor inputs.

We observe that machine learning algorithms influence largely in decision making process. But, there is need to understand the control system consequences for adapting the outcome of the machine learning algorithm. This proposal presents the detailed study on the influences of machine learning architectures and algorithms for modeling and control of nonlinear dynamical system.

Research Outcome:

· Knowledge on machine learning architectures (Support Vector Machines (SVMs), Conditional Random Field, supervised neural network)

· Understanding the constraints on applicability of ML architectures for nonlinear dynamical system

· Study on real time control of nonlinear dynamical system with ML algorithm in closed loop.

Study on application of Machine Learning Intelligence for Identification and Control of Nonlinear Dynamical Systems: Case study Robotic Manipulators

Divya Rao Ashok Kumar, Dr. Krishna Vijayraghavan

1.0 Abstract

The current machine learning algorithms influence largely in decision making process or applied to static systems (pattern recognition). But, there is need to understand the control system consequences in terms of system stability for adapting the machine learning algorithm in the closed loop control. This work presents the study on the influences of machine learning architectures and algorithms for modeling and control of nonlinear dynamical system. This research attempts to investigate the evolution of learning based controller design for nonlinear dynamical system specifically a robotic manipulator. The effort to understand the applicability of machine learning principles and adaptive control theory for identification and control through neural networks as the learning systems of the system being identified and controlled is studied.

Keywords: System identification and Control, adaptive control, Neural Networks, Robot Manipulators

Nomenclature

ANN	Artificial Neural Network
ARMA	Autoregressive Moving Average Model
ARX	Auto-Regressive with eXogenous
BPTT	Back Propagation Through Time
CAMC	Cerebellar model articulation controller
DARPA	Defense Advanced Research Projects Agency
DOF	Degrees of freedom
DSP	Digital Signal Processing
FMLP	Feedforward Multilayer Perceptron Model
FFN	Feedforward net
FARNN	Fully Automated Recurrent Neural Network
GRU	Gated Recurrent Units
GRBF	Gaussian Radial Basis Functions
GRNN	General Regression NN
GRRMLP	Globally Recurrent Multilayer Perceptron Model
LAGRs	Learning applies to ground robots
LIP	Linear In Parameter
LPAC	Linear perturbation adaptive control
LTI	Linear Time Invariant

LSTM	Long Short-Term Memory
MBRNN	Model Based Recurrent Neural Networks
MRAC	Model Reference Adaptive Control Approach
MIMO	Multi Input Multi Output
MNN	Multilayer Neural Network
MLP	Multilayer Perceptron
NN	Neural Networks
NARX	Nonlinear Auto-Regressive with eXogenous
PMDC	Permanent Magnet Direct Current
PE	Persistence of excitation
PD	Proportional Derivative
PID	Proportional Integral Derivative
RBF	Radial Basis Functions
RBFN	Radial Basis Functions Networks
RTRL	Real Time Recurrent Learning
RFMLP	Recurrent Feedforward Multilayer Perceptron Model
RMLP	Recurrent Multilayer Perceptron Model
RNN	Recurrent neural Network
RLS	Recursive Least Square
STAC	Self-Tuning Adaptive Control
SHL	Single Hidden Layer
SISO	Single input and single output
SLS	Supervised Learning System
TFFMLP	Teacher Forcing Feedforward Multilayer Perceptron Model
TFRMLP	Teacher Forcing Recurrent Multilayer Perceptron Model
ULN	Universal Learning Networks
WNN	Wavelet NN
WDEKF	Weight-Decoupled Extended Kalman filter

1.1 Introduction

Learning applies to ground robots (LAGRs) program by Defense Advanced Research Projects Agency (DARPA), RobotCub project, dynamic robot control project on ping-Pong playing robot at Bell labs, wide use of Unimate Puma 550 series, Google Brain Team executing reinforcement learning experiments on robot manipulators has shown the advent of large number of robot manipulators designed over the last half century as the standard platforms for research and development efforts and industrialization [1]. There is an abundance need for the robotic manipulators in the areas of space (space debris removal, On orbit servicing), robotic surgeries, prosthetics and manufacturing process. Smart robotics was considered to be the major driving factor for the migration of Industry 3.0 (i.e. mass manufacturing) to Industry 4.0 (i.e. customized or social manufacturing) [2]. Hence, many researchers have been applying the identification and control theory to robotic manipulators. A robot manipulator is said to possesses highly complex nonlinear robot dynamics. The uncertainties associated with manipulator which is capable of handling variable loads can be; apriori unknown mass properties of the load or its exact position in the end effector. The robot control problem is generally defined as the computation of joint torques or actuator inputs so that the joint motions (position and velocity vectors of the joints) closely track a desired trajectory [3]. As the manipulators in practical possess nonlinear dynamics, linearization of the system may simplify the mathematical representation of the system. However, the controller developed for the linear system cannot assure the desired performance when applied in practical. Hence, it is required to develop control strategies for stabilizing the nonlinear dynamical systems considering the nonlinearities of the system [4]. Many researchers have made a tremendous effort in ensuring the design of controllers to develop accurate robot manipulators used for numerous applications. This research attempts to investigate the evolution of learning based controller design for nonlinear dynamical system specifically a robotic manipulator.

The learning-based systems had an evolutionary beginning in 1962, when American scientist F. Rosenblatt had introduced the concept of learning in his book Principles of Neurodynamics. He dealt with physico-mathematical models for the psychological functioning of the brain as expressed in terms of the known principles of neuroanatomy and physiology. He was successful in constructing a number of computer oriented realizations of brain models with formal training methods (reviewed by A. A. Mullin [5]). His work was a source of inspiration for a new approach to the simulation of the functions of the brain. Additionally, the work in probabilistic iterative methods for stochastic approximation and problems of identification, filtering and control to be solved with adaptive techniques was inspired by the book by Ya. Z. Tsypkin’s, Adaptation and Learning in automatic Systems (reviewed by B. L. Deekshatulu [6] and K. S. Narendra[7]). Inspired by learning-based systems, Arimoto. et.al [8] proposed the three models which uses feedforward signals for learning of specific tasks without explicitly updating the system dynamics. The interactive learning control schemes was applied for linear and nonlinear dynamical systems, specifically for controlling the robotic manipulators. Slotine et al. derived the adaptive robot control algorithm with the unknown manipulator and payload parameters being estimated online [9]. Many experiments conducted by K J Astrom and others in 80’s designed adaptive control laws assuming the order of the system is known a priori, this assumption was considered unrealistic due to the presence of unmodelled dynamics [10]. The adaptive control techniques were designed to handle the anomaly situations occurred due to interaction between disturbances and unmodeled dynamics [4], [9], [11]. The application of conventional model based adaptive and robust control techniques for nonlinear systems with parametric uncertainties is still an ongoing research. However, the system model for the development of the adaptive control law and accuracy of the model impacting the overall performance of the nonlinear dynamical system to achieve desired trajectory tracking is the primary objective to be addressed. The model free controller designs were the possibility to satisfy the primary objective for the systems with nonlinearities and unmodelled dynamics [12], [13]. However, the stability proofs of controllers to satisfy the insensitivity to parameter uncertainties and insensitivity to unmodelled dynamics with faster convergence rates and less computational load is the objective of research for model free controllers.

However, learning based model free controllers design were attractive because it does not require a structural or parameterized model of the system dynamics. The functions of biological process are adapted to learn about the system to improve the performance. Model free learning controllers was tried to be implemented with fuzzy logic control, neural networks (NN) and genetic algorithm. It was generally understood that NNs provide an elegant extension of the adaptive control techniques. However, genetic algorithm can be used to optimize control parameters of NNs. The NNs in controls was initially proposed by Werbos, Narendra and Slotine [12]–[15][16]. The NN is formed from compositions and superpositions of a single, simple nonlinear activation or response function [17][18]. The output of the network is the value of the function that results from that composition and superposition of the nonlinearities. Norgaard et al., [19] provided the MATLAB based tools for system identification and control with NNs.

This study is focused on the development of control laws for nonlinear dynamical systems adapting biological signal processing-based learning methods usually referred as NNs and its recent developments on integrating the conventional control strategies with learning-based approximation models [13], [20]–[34]. Training the networks is considered as the vital component for design of stable controller. The most widely used network training algorithm, back-propagation was designed and published by Paul Werbos [13]. With conventional back propagation through time (BPTT) [35], [36] or real time recurrent learning (RTRL) the development of training large scale complex NNs has become very slow or even stagnated due to computation requirement and vanishing gradient problem. The resutls firstly from the computational powers needed bytraining NNs by computing their weights through back propagation, especially when the networks have multiple layers and vast numbers of hidden nodes. Secondly, the vanishing gradient problem: the gradient of errors will vanish gradually through the back-propagation process. However, vanishing gradient problem is a problem of concern as the processing units with high computational powers are available. The vanishing gradient problem was firstly addressed by Hochreiter [37]. This led to the numerous works in the area of deep learning with NN structures with reasonable training speeds. The three classes of NNs such as multilayer neural networks (MNN), recurrent neural networks (RNN) and recently Long Short term memory network (LSTM) are used in many applications[38][39][36]. MNN, RNN and RFBN's have been used in identification and control of static and dynamic nonlinear systems [22], [23], [25], [26], [40], [41]. Additionally, recurrent networks are well known for its application in associative memories for solutions of time series optimization problems.

The adaptive control methods have focused on the process of controlling engineering systems to solve the problem of tracking control parameters in the presence of uncertainties in system models, changes in the environment, and other unforeseen variations. The approach used for accomplishing tracking is to learn the underlying parameters through an online estimation algorithm. Stability theory is employed for enabling guarantees for the evolution of the control parameters and convergence of tracking errors to zero. On the other hand, machine learning has objectives focused on the processes improvisation through knowledge and the principles that govern learning systems. Both Adaptive control theory and principles of the machine learning go hand in hand for accomplishing this process of automatic improvement of the system through learning parameters of the model using input-output data and application of the optimization theory to reduce the approximation and control errors at optimal rates often inspired by gradient descent [42]. The effort to understand the applicability of machine learning principles and adaptive control theory for identification and control through neural networks as the learning systems of the system being identified and controlled is studied in future sections of the report.

1.2 Adaptive Control systems for robotic manipulators

Arimoto et.al [8] proposed three data based interactive learning control schemes for linear and nonlinear dynamical systems in 1985 and applied the work for controlling the nonlinear robotic manipulators that had a serial link manipulators. His work considers the use of feedforward signals for learning of specific tasks without explicitly updating the system dynamics. However, in early 80's researchers had presented the feedforward control approach for direct drive arm for system with slow and fast movements of the links. Atkenson et al. [11] presented experimental results of using an estimated dynamic model of the manipulator for dynamic feedforward compensation for trajectory control of serial link direct drive arm with fast movements. The feedforward control was affected by the accuracy of the manipulator dynamic model and the paper noted that the unmodelled dynamics resulted in poor trajectory tracking performance. Atkenson et al. assumed the inertia of the link to be much smaller that other links of the arm, the unmodelled dynamics such as the non-ideal dynamics of the motor and the friction of the bearings were dominated by the modelled rigid model dynamics clearly indicating the need of complete modelling of the link 3 to improve trajectory following. The feedforward control demands for the precise estimation of the nonlinear multi-link dynamics parameters such as inertia matrix, vector of Coriolis and centripetal terms, gravity vector and friction terms of each link. The estimation of the parameters involved linearizing the Newton-Euler equations for an n degree of freedom (DOF) manipulator in terms of unknown inertial parameters. Atkenson et al. used least squares algorithms to compute the estimates of the multi-link dynamics parameters. The accuracy of the estimated parameters was verified by comparing the measured joint torques to the torques computed from the estimated parameters. This work clearly indicated the effect of unmodelled dynamics and accuracy of estimated feedforward parameters on the precision of trajectory control of nonlinear dynamical system.

The parallel experiments by Astrom [4] on adaptive controller performance under the effect of anomaly such as interaction between disturbances and unmodelled dynamics provided the influence of model reference adaptive control system on single input single output (SISO) continuous time system. Researchers has provided the stability proofs for simple adaptive algorithms considering the order of the plant is known. However, many pointed that the assumption on order of the plant is highly unrealistic due to the presence of unmodelled dynamics. The experiments by Astrom [4] indicated that the equilibrium is non-unique, and that small disturbances may make the move the equilibrium until the system becomes unstable. The author have noted that while the equilibrium is the globally stable for the nominal plant, stability cannot be guaranteed in the presence of unmodeled dynamics The need was to determine the adjustable feedback gain parameters of the SISO system. The rate of change of the parameters was adjusted with adaptive gain and the proper gain adjustments was possible with bounded input and output of the nonlinear SISO system and the type of excitation (step or sinusoidal) applied to the system. The relationship between the adaptable gains and the stability boundaries in presence of unmodeled dynamics indicate the importance of initial values of the adaptable gains and its convergence towards equilibrium set (not point) to provide stable closed loop system. However, the performance of adaptable gains under the effect of disturbances is expected to drift. When measurement noise was included as disturbance in the system, the adjustable gains tend to oscillate in the neighborhood of the equilibrium set, hence leading to the drift of the adaptable gains i.e., closed loop gain increases as long as there are disturbances affecting the system. His experiment provided a thumb rule indicating the crude estimate of the drift rate of the adaptable gains drift to infinity at a rate proportional to $\sqrt[3]{t}$, t being the time of execution and indicated that the generic case of unmodelled dynamics the increased adaptable gain will provide unstable closed loop system. K J Astrom provides an insight that local stability is related to persistent excitation because of guaranteed unique equilibrium, however to achieve a stable equilibrium the necessary condition is that the excitation is in the proper frequency range where the range depends on the model developed and the true dynamics of the system being modelled.

Further, the survey conducted by Hisa [ref], resulted in categorizing control schemes as the fixed or non-adaptive and adaptive control and further categorizing adaptive control systems into model-reference adaptive control, self-tuning adaptive control and linear perturbation adaptive control applied to robot manipulators. Hisa highlighted the necessity to consider the payload parameters; actuator dynamics (robot joints drivers); nonlinear motor and friction torques in the robot manipulator dynamics. If a robot manipulator such as three link direct drive arm is considered the robot control is required to compute the joint torques or the actuator inputs such that the joint motions closely track the desired trajectories. The classical controllers controlling each joint individually fail to consider the nonlinearities of the process and the dynamic coupling between the joint motions. It is addressed that the high accuracy positioning of the end effectors of the manipulators in the inertial space is achieved by considering the nonlinearities and joint couplings in controller design. However, fixed/ non-adaptive controllers can be the best choice if exact knowledge and explicit use of the complex system dynamics and system parameters are available. Further, considering the performance degradation leading to instability due to system model uncertainties, the adaptive control schemes with automatic compensation of the uncertainties by adjustable gains was highlighted. Hisa, listed the considerable uncertainties in all robot dynamic models which are either impossible to know precisely or some can vary unpredictably viz. mass and inertia, variable payloads, elasticities and backlashes of gear trains. In addition to the uncertainties, the actuator limitations such as saturation, minimum response and loadings also be accounted in control design. Hisa, clearly pointed out the objectives for the design of robot controllers followed by the control community working towards adaptive schemes for robot control; which are stated as insensitivity to parameter uncertainties, insensitivity to unknown payload variations, decoupled joint response and low demand for online computations. It is required to further analyze if these objectives were able to be satisfy one or more objectives by the researchers.

- **Model reference adaptive control**

The adaptation algorithm of MARC approach is designed based on asymptotic stability requirement, which are analyzed using Lyapunov stability criterion and hyperstability criterion. The Lyapunov MARC design approach, had time varying adjustable feedback and feedforward gain matrices derived from the state errors, with the goal of asymptotic zeroing of the error and minimization of a performance index based on Lyapunov stability criterion [43]. Durrant-Whyte, demonstrated the insensitivity of his algorithm to load variations and uncertain dynamics knowledge. His work formulated the Pennsylvania Anthropomorphic Robot Manipulator as an open kinematic chain actuated by DC servo motors. The servo motors were represented by a second order model with input voltage, torque constant, terminal resistance, actuator-link gear ratio, actuator inertia, actuator damping, back EMF, link side displacement and the random motor disturbance at about 2% of stall torques, including friction and dynamic effects to the link side as inputs to the model. The quasi linear formulation of the manipulator feedforward block was derived for n joints, including all loads and actuator dynamics which included non-linear motor, friction torques and torques associated with the motion of the link structure. His simulations were conducted for unloaded and loaded (5 kg load at the end effector) motion, the adaptive algorithm was capable of automatic selection of reference model and initial gains. The author proves the convergence of the manipulator's position tracking the desired position, satisfying the objectives such as insensitivity to parameter uncertainties, insensitivity to unknown payload variations, decoupled joint response and low demand for online computations. The work also provided the insight that algorithm is consuming less computation time thus allowing for high sampling rates (ARM Architecture: processing unit). However, both unloaded and loaded motion converged within 2 secs with no significant convergence time difference between the two cases. The results do not provide any insight on time of convergence requirements. It is also essential to understand the role of convergence times in the adaptive control schemes of robotic manipulators.

Lim et al. developed different controllers for three joint revolute robot manipulator systems (nonlinear time varying) based on the general adaptive control theory using Lyapunov direct method. This work aimed at improving the transient response and convergence speed [44]. The large state error and/or oscillation occurring during transient time is addressed by introducing approximate auxiliary control input. The reason of choosing adaptive control over robust and decoupled control methods related to the reduced computing time of former methods in generation of nominal torques of the joints. Lim et al. observed the need for decoupling the dynamic equations of the manipulator and implemented control schemes with decoupled model matching. However, his simulation results showed better transient response compared to the controller design presented by Durrant-Whyte. His research indicated the need for advancements in computation powers to develop robust control schemes for manipulator design with real time online computation of dynamic coefficients which were used in the nominal feedback controller gains to enhance the overall performance of the manipulator.

Further researchers classified the adaptive control into direct and indirect methods. K. S. Narendra et.al [45], initially defined the classification of two methods of adaptive control problem such as indirect and direct control methods for linear time invariant plants. The purpose of his work was to understand that if there exists a controller structure (direct/indirect) which can generate the appropriate control input, additionally understanding the generation of adaptive laws and stability proofs that the control parameters with arbitrary initial conditions can converge to desired values with required accuracy. If in the problem the plant model (parameters and /or state variables of the unknown plant) is identified/estimated online and if these in turn are used to adjust the parameters of the controller, the method is defined as indirect adaptive control method (self-tuning regulators). If in a problem there exist a model (number of poles > number of zeros) of the desired performance, the control parameters are adjusted so that the error between the plant output and reference model output tends to zero asymptotically such method is defined as direct adaptive control method. Additionally, identification error in indirect control and control error in direct control are used to update the control parameters [46]. However, for both methods the fact is that the plant is essentially a black box and only the input is accessible for control purposes. K.S. Narendra's work clearly defined a building block to classify the adaptive control law development into two different methods, but it was essential to project the work in the area of robotic manipulators which are highly nonlinear by nature of dynamics. Researchers considered the indirect adaptive control which requires parameter identification as difficult to achieve for robotic manipulators because of large number of parameters. Hence, the direct method was a good option as it provides robot controller with relatively simple adaptive laws, computed at reasonable cost. The parameter adaptation mechanism was developed using Lyapunov stability or Popov hyperstability that guarantees overall system stability.

Hyperstability MRAC design approach discussed in Hisa's survey provided an alternating way of designing an adaptive control law using stability condition. In comparison of adjustable feedback and feedforward gain matrices with Lyapunov design the derived gain matrices using hyperstability theory were divided into an adjustable part and a fixed part. However, the qualitative difference in performance were not recorded. The simplified MRAC design presented by Dubowsky and DesForges addressed the design of adaptation laws by gradient search techniques which resulted in very simple adaptive laws, however the global stability was not always guaranteed. Another MRAC scheme developed by Horwitz and Tomizuka is applied to adaptively compensate the nonlinearities and decoupling the joint motions. The joint dynamics were modelled by a double integrator and used simple fixed Proportional Integral Derivative (PID) for feedback control. The system also had manipulator model without gravity force effects. The control law with adaptively estimated parameters ensured adaptive system stability. The design of the adjustment algorithm for adaptively estimated model parameters is investigated using hyperstability technique[3].

- **Self-tuning adaptive control**

Another popular approach for robot controller design is the self-tuning adaptive control (STAC) [10]. If the plant is represented by linear discrete time models for digital implementation, the adaptive controller designed for such plant is defined as STAC. Yuh presented a discrete model reference adaptive control to the flexible link of robot mechanisms. The controller was developed in the perspective to overcome the inaccuracy due to structural vibrations of the mechanisms. The results were presented for nonlinear flexible link model equations. His work was restricted to control of the angle of the joint, additionally unlike other researchers work mentioned above, he added process noise to the dynamic equation of the manipulator. His work lacked to provide the stability proofs for the adaptive controller developed. The effect of unmodelled flexible dynamics with candidate mode frequencies was derived using the Bernoulli-Euler beam equations with appropriate boundary conditions, assuming the links are made of Aluminum. However, many researchers recommended applying STAC on slowly time varying plants. The discrete time plant model must be established through the system identification technique using sampled input-output data of robot model dynamics for STAC problem. The modelling of the robot for STAC technique can be a linear but time varying process and/or m^{th} order scalar difference equation of the Autoregressive Moving Average Model (ARMA) type process. Since the model parameters are in general time varying, they are recursively estimated online. The parameter estimation from the input-output data measurements are performed using least squares estimation algorithm. The robot controllers designed with STAC technique also was effectively implemented for joint velocity control instead of position control, this led to simpler adaptation algorithm as the controller parameters were directly related to the system model parameters [3].

- **Linear perturbation adaptive control**

The yet another approach for robot controller design is the linear perturbation adaptive control (LPAC). If a nominal robot model is available, instead of assuming the robot dynamic equation parameters are completely unknown. Then the nominal values of the model parameters are determined using inertia and gravity loading functions. With this type of plant model, the adaptive control is designed using linearized perturbed state equation formed using Taylor series and neglecting the higher order terms with the control objective to determine the control error to make system asymptotically stable [3]. Previously, Arimoto et al. had developed the adaptive control design techniques for the trajectory control using the nominal plant model which included the approximate inertial and gravity leadings. They used the MRAC Lyapunov design technique for deriving the gain matrices of the adaptation law [47]. LPAC shows advantage in using the simple plant model but the adaptive controller with perturbation model parameters being estimated online slows down the adaption rate in real time applications, hence the simpler decoupled adaptive controllers are much preferred for robotic manipulators. Additionally, Hisa's survey indicated that the selection of number of parameters characterizing the controller performance in the control design is proportional to the computational load on the processing unit.

- **Advanced Adaptive controllers**

The advanced adaptive controllers are expected to provide the elegant adaptive control laws with the stability proofs with the reduced computational load. J.J.E. Slotine et al. attempts to derive computationally simple adaptive robot control algorithm which consists of Proportional and Derivative (PD) feedback controller and full dynamics feedforward compensation (inertial, centripetal and Coriolis, and gravitational torques), with the unknown manipulator and payload parameters being estimated online [9]. His work analyzed the dependence of the system dynamics on the unknown parameters can be made linear in terms of suitably selected set of robot and load parameters for a two-link manipulator. The controller was designed considering the known desired trajectory (position, velocity and acceleration of joints), with some or all manipulator parameters being unknown. The control objective was to produce the actuator torques and development of estimation law for the unknown parameters such that the manipulator tracks the desired trajectory after an initial adaptation process. A simple globally stable adaptive controller for position and velocity control of the manipulator was designed from a Lyapunov stability analysis. However, the adaptive controller resulted in a non-zero position errors. The positional errors were forced to converge by restricting the error residuals to lie on a sliding surface.

Slotine et al. discussed the implementation which highlighted the detailed design considerations of the controller. Firstly, even though the convergence of the trajectory tracking was guaranteed, it was observed that the parameter estimates themselves do not necessarily converge to their exact values. Hence to guarantee parameter convergence, the desired trajectory must be "sufficiently rich" so that only the true set of parameters can yield exact tracking. The practical implementation of the adaptive controller as the error terms vary much faster than the dynamic coefficients matrices, the controller parameters are updated at slower rate compared to reference trajectory generation. The choice of the adaptation gain matrix is generally such that the adaptation process is slower than the control bandwidth. He even introduced a recursive Newton-Euler method as an alternative way of implementing the control and adaptation laws, which usually needs a number of operations proportional to the number of links. Additionally, they simplify the adaptation algorithm by not explicitly estimating all unknown parameters. If some parameters have relatively minor importance in the dynamics, then they design the controller robust to the uncertainty on these parameters rather than explicitly estimating then online. A sliding control term is incorporated into the torque input to account for of disturbances and the effects of uncertainties on the non-estimated parameters . Additionally, the adaptation algorithm avoided long term drift of the estimated parameters and undesirable control chattering. The simulation study was performed for a desired joint trajectory (with a 5^{th} order polynomials) using PD Controller, PD Controller and full dynamics feedforward compensation and adaptive controller with sliding control term, the position errors of the adaptive controller is minimum with the smaller magnitudes of actuator torques. But still the need for specific choices of the adaptation gain matrix that yield optimal convergence rates while still avoiding the excitation of high-frequency unmodelled dynamics is required.

Rokui et al. provided indirect adaptive feedback linearized controller for a single-link flexible manipulator modeled in discrete time [48]. The authors used the discrete time model to obtain the associated internal and zero dynamics. The unknown payload is identified by using a new regressor form of the system dynamics and the multi-output recursive least square (RLS) algorithm. The output re-definition strategy is employed to guarantee that the map between the hub and the new output to be minimum phase. Stability analysis of the adaptive controlled closed loop system is guaranteed by applying Lyapunov analysis.

The above study clearly indicated the need for the system model for the development of the adaptive control law and accuracy of the model impacting the overall performance of the nonlinear dynamical system. The adaptive controller was derived both in continuous and discrete domain. However, the discrete time controllers are not extensively developed with the stability proofs. The need of controllers to satisfy the insensitivity to parameter uncertainties and insensitivity to unmodelled dynamics with faster convergence rates and less computational load is being studied.

1.3 Learning based controllers for robotic manipulators

In 1962, Rosenblatt [ref], Tsypkin [ref] and other literature [49] introduced the concept of *learning* and adaptation. This work inspired many to work in learning control system.

The control objectives of NN-based controller design are similar to adaptive controller discussed by researchers mentioned above. The objectives are ability for adaptation in real time to new environments (eg. payload parameters) without instability; ability to handle nonlinearities and noise, ability to plan or optimize over time (as required with complex tasks); parallel control of multiple actuators, adaptable for parallel computing; coupling between a slower controller and faster subordinate system, similar to the implementation presented by [9] for adaptive controllers.

Werbos [12], discussed application of the artificial neural networks (ANN) for control and system identification of systems for robotic systems. ANN based adaptive systems are classified into supervised and unsupervised learning systems. In supervised learning, the ANN performs a mapping from a set of independent variables (inputs) to a set of dependent variables (outputs/targets). The ANN contains a set of connection weights, which needs to be estimated recursively so as to minimize the squared error between the ANN outputs and desired outputs outside of the training set. The adaptive controller design implements similar approach for adaptation laws developed by STAC problem approach mentioned by [10]. The problem of minimizing square error from the adaptation of parameter estimates in real time is very different to the static database. The immediate step is to calculate the rate of change of the error with respect to all parameters. Normally, the procedure is named as backpropagation [13] which essentially calculates all the required derivatives at every instant; applicable for any system not restrictively ANN which has a differentiable network or model. However, speeding up the rate at which backpropagation is repeated can minimize the error faster but asks for parallel implementation and directly influence the computational load leading to the requirement of high processing platforms [50]. The difference between $O(N^2)$ cost and $O(N)$ cost (N being the number of samples) decides the network sizes and its computational platform requirements. Supervised learning is identical to the classic problem of nonlinear regression. SLS are commonly used to solve classification problems, by letting the inputs be the patterns to be classified and the desired outputs be the correct classifications [12] [51]. However, in reinforcement learning systems the system is given a reward for its output without being explicitly told the desired output [52]. By contrast, in unsupervised learning, the systems “learns” by observing their environment over time where there exist no desired output/s. This scheme is typically designed to act as associative or content addressable memories, as feature detectors or forecasting networks [12][51].

Werbos presented five approaches to ANN design for control of robots. (I) The NN learn from the desired actions as a function of the sensor inputs, it learns from the mapping from the sensor reading to the desired outputs, named as supervised control system. (II) The robot learns to follow trajectory planned by a planning system. The problem is to learn the mapping from spatial coordinates back to the actuator signals (joint angles control), along with SLS to learn the mapping. The method is named is direct inverse control. (III) A NN learns to minimize or maximize any performance criterion, till a model of the system to be controlled is developed. The method is popularly known as backpropagation through time (BPTT). The application of BBPT to adapt the control network for robot arm (developed at MIT and Japan’s Bell Labs). This involved defining the performance criterion representing the deviation from the desired trajectory to the actual trajectory, along with a measure of smoothness of motion. Followed by adapting a network to describe the dynamics of the arm. (IV) Neural adaptive control, a conventional adaptive control replaced by NNs replacing some of the usual linear mappings[14]. (V) Adaptive critics system (a complex subject), class of designs which tries to perform optimization over time[53]. Allowing explicitly for the possibility of noise, and also allowing true real-time learning. The adaptive critic designs can be defined as designs which try to approximate Bellman equation of dynamic programming. The search for good approximations to dynamic programming. Adaptive critics involve a 2- network design, critic network and action network[52], [54]–[56][2].

F.J.Pineda defined NN approach as a paradigm for computation in which the traditional paradigm of a finite state machine performing sequential instructions in a discrete state space is replaced with the paradigm of a dynamical system, in a discrete or continuous state space, which evolves under the control of certain class of dynamics [57]. The dynamics of the NN has three salient features. Firstly, the dynamical system has many degrees of freedom. As per Pineda, most simulations of these systems were limited to 10^5 neurons while human brain has 10^{11}neurons. The activity level and the time derivative of the activity of the neurons are the coordinates in the phase space/state space of the system. The second feature is nonlinearity, which is essential to build a computing machine. The third feature is dissipative, which is characterized by the convergence of the state space volume onto a manifold of lower dimensionality as time increases. Pineda, stated that systems whose flow exhibits the property of global asymptotic stability (systems achieve steady state for any choice of initial condition) play a particularly important role in NN modelling.

If NNs are used the work is to determine an adaptive algorithm or rule to adjust the parameters of the network based on the given set of input-output pairs. If the weights of the networks are considered as elements of a parameter vector, the learning process involves the determination of the new parameter vector which optimizes a performance function based on the output error. Backpropagation is the most commonly used method for determining the parameter vector which optimizes a performance function/ cost function. The adjustment of parameters is carried out by determining the negative gradient of cost function in parameter space (cost function w.r.t parameter), the procedure commonly followed is to adjust it at every instant based on the error at that instant and a small step size [13], [14], [58].

However, before controlling the system we need to understand the system being controlled. The model of the system can be deterministic or stochastic. The quality of the NN controller depends on the quality of the system model, like adaptive controllers. The attempt to understand the development of NN based model when an explicit model of the system is unavailable / difficult to arrive at. This problem is called as system identification, the NN developed is named as identifier network[54]. The two-part networks are applied to many applications as identifier networks, major disadvantages are a stochastic model with uncorrelated noise which cannot approximate even vector-ARMA process and robustness over time. The problem of allocating noise in maximum likelihood context is tough, as the requirement is to approximate true nonlinear filtering. The encoders and decoders schemes and Competitive learning schemes are suggested for nonlinear maximum likelihood case (stochastic process approximation)[59].

The identification and control of dynamical systems using neural networks (NNs) by K.S. Narendra et al. was an initial advancement towards development of identification as well as controller structures using NNs for adaptive control of unknown nonlinear dynamical systems. The work presented is considered as the significant research work in the area of identification and control of nonlinear dynamical system providing an elegant extension to the adaptive control systems theory [14]. The paper has more importance due to the consideration of discrete time system representation by difference equations for easy implementation. The controller and identification model were developed using multilayer and recurrent networks interconnected in novel configurations. Both static and dynamic back-propagation methods are used for adaptation of parameters. The paper presented a perspective method for the dynamic adjustment of the parameters based on back propagation; named as dynamic backpropagation.

K.S. Narendra et al. provided a detailed insight on identification of the system which is the basis for understanding the future research in this area. His research identified that the conventional system identification started with system characterization where the problem of characterization is concerned with the mathematical representation of a system. The problem involves mapping of input space to output space using the model of the system expressed as operator which belongs to the larger class of operators. The problem of identification is to approximately identify the operator which is close enough to the model of the system expressed as operator as well as which belongs to the approximately identified larger class of operators. In static systems the input space and output space are subsets of real space of different dimensions. However, the dynamical systems are generally assumed to be bounded Lebesgue integral functions on the interval. As per, K.S. Narendra et al. the operator is defined implicitly by the specified input-output pairs. The choice of the approximate model of the system expressed as operator as well as specific method used to determine its approximate larger class of operators depends on factors related to accuracy desired as well as the analytical tractability. These include

- the adequacy of the model of approximate larger class of operators represent larger class of operators,
- its simplicity,

- the ease with which it can be identified,
- how readily it can be extended if it does not satisfy specifications and
- finally, the approximate larger class of operators chosen to be used online or offline.

In practical applications many of these decisions depends on the prior information that is available concerning the plant to be identified.

K.S. Narendra et al. described the problem of the pattern recognition is a typical example of identification of static systems. The compact input set of pattern vectors are mapped onto the output class of elements in the output space by a decision function expressed as larger class of operators. In dynamical systems, the larger class of operators defining a given plant is implicitly defined by the input-output pairs as function of bounded parameter time. However, in both cases the objective is to determine the larger class of operators so that normalized identification error is within a permissible bound.

The famous approximation theorem of Weierstrass states that any function in the space of the continuous real valued functions can be approximated arbitrary closely by a polynomial. K.S. Narendra et al. indicated that Weierstrass's theorem and its generalization to multiple dimensions finds wide application in the approximation of continuous functions using polynomials. A generalization of Weierstrass's theorem due to Stone, called the Stone- Weierstrass's theorem can be used as the starting point for all the approximation procedures for dynamical systems[60]. Using Stone-Weierstrass theorem it can be shown that a given nonlinear functional under certain conditions can be represented by a corresponding series such as the Volterrra series or the Wiener Series. K.S. Narendra et al. claims that very few series found wide applications in the identification of large classes of practical nonlinear systems. Hence, K.S. Narendra et al. worked on the representations which permit online identification and control of dynamic systems in terms of finite dimensional nonlinear difference or differential (discrete or continuous) equations.

K.S. Narendra et al. describes the identification problem of the unknown system considering input and output of a time invariant, causal discrete time dynamical plant. Where, he considers the input is uniformly bounded function of time and the plant is assumed to be stable with known parameterization but with unknown values of the parameters. The objective was to construct a suitable identification model which when subjected to the same input as the plant, produces an output; such that so that normalized identification error is within a permissible bound. K.S. Narendra et al. defines parallel identification model and series parallel identification model which might help us in understanding the future implementations. Parallel identification model is the output of the identification model at an instant is a linear combination of its past values as well as those of the input. Series parallel identification model is a linear combination of the past values of the input and output of the plant. Series parallel model is preferred for the generation of the stable adaptive laws.

K.S. Narendra et al. describes the control problem of the unknown system as the dynamical system with one or more variables are kept within prescribed limits, with the known state transition matrix and output matrix, the problem of control is to design a controller which generates the desired control input based on all the information available at the instant. For the control of nonlinear plants even when the state transition matrix is known, and the state vector is accessible, the determination of control vector for the plant to have a desired trajectory is considered to be difficult. Hence, K.S. Narendra et al. assumed the existence of suitable inverse operators for the generation of the control input. If a controller structure is assumed to generate the input control vector, further assumptions were made to assure the existence of a constant control parameter vector to achieve the desired objective. The choice of structures for identifiers and controllers in the nonlinear case was motivated by the linear models.

K. S. Narendra et al. used two classes of neural networks; multilayer neural networks (MNN) and recurrent neural networks (RNN) and indicated that a unified treatment of the two, which might deal with more complex systems. The MNN (versatile nonlinear maps) with an input layer, two hidden layers and an output layer, with three weight matrices and a diagonal nonlinear operator with identical sigmoidal (activation function) elements following each of the weight matrices is used[61]. The weights of the network are adjusted using adaptive rules to minimize a suitable function of the error between the output of the network and a desired output. K. S. Narendra et al. mentioned that a discontinuous mapping such as a nearest neighbor rule can be used at the last stage to map the inputs sets into points in a range space corresponding to output classes. Recurrent networks (associative memory) was introduced in the works of Hopfield [62]. The Hopfield network consists of a single layer network included in feedback configuration with a time delay representing a discrete time dynamical system. For recurrent networks, the set of initial conditions in the neighborhood of initial state which can converge to the same equilibrium state is then identified with the state. Pineda, also stated that Hopfield model guarantee to be globally asymptotically stable, as it can minimize an energy function [57].

K. S. Narendra et al. presented static and dynamic back propagation methods for the developed NNs [63]. Pineda, claimed that for RNNs, the identification of stable fixed points with computational objects (memories), is one of the fundamental requirement which is satisfied by the control of the locations of the fixed points of the NNs[57]. Hence, a learning algorithm is a rule or dynamical equation which changes the locations of fixed points to encode information. Pineda also suggested the use of gradient descent to minimize the function of system parameters and defined the dynamics of the network based on coupled differential equations having logistic activation functions. The research by Pineda, addressed forward propagation was widely used in feedforward networks and it used δ rule as a learning rule as well as its adaptability to recurrent networks. Recurrent back-propagation algorithm was developed to ensure substantial self-excitation leading to oscillatory solutions as compared to feedforward networks with constant excitation where solution exponentially decaying to constants[57]. Pineda also tries to find a learning algorithm to adjust the parameters such that the fixed initial state and a given set of input values result in a fixed point, whose components along the output units have a desired set of values. The necessary condition for the learning algorithm is to reach steady state and achieve global asymptotic stability of system. Hopfield's equations were globally asymptotically stable considering symmetric parameter space with zero diagonal elements and the same was proved using Lyapunov functions. A general theorem concerning the stability of the networks with symmetric weights was given by Cohen and Grossberg[49][64]. Philip Tai et.al. presented a detailed survey on the applications of neural networks to control systems [65]. His work highlights the algorithms and techniques developed by several researchers using NNs. The use of NNs for system identification, adaptive control, modeling of chemical processes, optimization, fault detection and control of robotic manipulators is described.

K. S. Narendra et al. state that for identification and control problem the inputs rather than initial conditions represent the patterns to be classified. Hence, recurrent networks with or without constant inputs are nonlinear dynamical systems and the asymptotic behavior of the such systems depends both on the initial conditions as well as the specific input used. In both cases, the asymptotic behavior depends on the nonlinear map (with activation functions) represented by the NN used in the feedback loop. The two-layer recurrent networks and more general forms of recurrent networks are constructed by including multilayer networks in the feedback loop of recurrent networks [66][67]. If a time delay element is added in the feedback path of the multilayer NN layer results in a recurrent network.

The basic building blocks of the general recurrent network is delay, summation and the nonlinear operator (with activation functions). In continuous time networks, the delay operator is replaced by an integrator or multiplication by a constant. K. S. Narendra et al. presented four generalized NNs constructed by connecting the transfer matrices of linear systems and nonlinear operators in cascade and feedback configurations, such that these generalized NNs form the building blocks for the more complex NN systems for identification and control of the class of nonlinear dynamical systems. The research of Kurt Hornik et al. showed using the Stone– Weiersrass theorem, that a layer network with an arbitrarily large number of nodes in the hidden layer can approximate any continuous functions [68]. This research motivated K. S. Narendra et al. that the class of generalized networks defined can deal with large class of generalized networks. Further, K.S. Narendra's work on representing a nonlinear dynamic plant with recurrent networks containing multilayer networks (four generalized NNs) represents an important area of research for the study of the stability properties.

K. S. Narendra et al. introduced the four models in discrete time to characterize plants using nonlinear difference equations for the representation of SISO plants which can be generalized to the multivariable case, where generalized NNs having multilayer NNs can be constructed to approximate these models. The properties of the models are described in the paper. The assumption that the functions of the models belong to a known class of functions generated by a network containing N layers, so that the plant can be represented by any one of the four generalized NNs or by a combination of generalized NNs. Thus, motivating the choice of the identification models and allows the statement of well posed identification problems. The identification models have the same structure as the plant but contain NNs with adjustable parameters. It is assumed that the plants being characterized has bounded outputs for the class of permissible inputs. They earlier mentioned the two categories of identification models such as parallel model and series parallel model. K. S. Narendra et al. claims that parallel identification model could not guarantee that the parameters will converge, or the output error will tend to zero. They suggest that series-parallel model of identification model (the output of the plant is fed back into the identification model) with static back propagation can be used to adjust the parameters reducing the computational overhead. The nonlinear plant identification using one of the four characterization models were presented. The NNs for series parallel identification model used static backpropagation was applied to five problems and the sixth problem used dynamic back propagation. The plant model was developed from one of the four characterization models presented in the paper. The choice of number of NN is proportional to number of unknown functions. The performance criterion of the NN's was to minimize the squared identification error; identification error defined as the error between the identification model and the characterized plant model asymptotically. The general NN structure is represented as $\mathfrak{N}^{N}_{i_1 i_2,\ldots\ldots,i_{N+1}}$: N: layers, i_1 :inputs, i_{N+1}: outputs, (N-1): hidden layers. The identification models had 4-layer networks (N=3), i_1 :inputs were dependent on the inputs of the unknown function and i_{N+1}: outputs were dependent on the outputs of the unknown function, 2 hidden layers with 20 and 10 nodes. The static propagation was performed with 0.25 step size and the dynamic propagation was performed with 0.01 step size. Hence, the methods for identifying nonlinear plants using delayed values of both plant input and output with adequate accuracy was presented by the authors.

After identification K. S. Narendra et al. presented the methodology for the adaptive control of Nonlinear systems using NNs. The controller model presented by authors shows that the controller whose output is the controller input to the plant and whose inputs are delayed values of the plant input and output. As the methods for directly adjusting the control parameters based on the output error (between the plant and the reference model outputs) are not available for the unknown nonlinear plant. Hence the adaptive control of nonlinear plants is arrived from the indirect control methods [45]. It is suggested by authors, that for offline control, identification of the model is performed initially, once the weights in the identification model are adjusted and had been identified to the desired level of accuracy, the control action can be initiated so that the output of the plant follows the output of a stable reference model. For online control, identification and control is performed simultaneously at same/different discrete time intervals, over which the identification and control parameters are to be updated have to be chosen properly. The simulations suggested that for stable and efficient online control, the identification must be sufficiently accurate before control action is initiated and hence the discrete time intervals for the parameters update for NN Controller and NN Identifier should be chosen appropriately. It is advised to update the parameters of NN designed for controller at slower rate compared to the parameter update of NN used for identification. The NNs structure specifications designed for controller and learning of NN controllers are not presented in the paper. Paper also does not provide the justification towards the selection of number of nodes for the hidden layers.

Chu et al. present two approaches for utilization of NNs in identification of dynamical systems, (I)A Hopfield network is used to implement a least squares estimation for time varying and time invariant systems and (II) Utilizes a set of orthogonal basis function and Fourier analysis to construct a dynamic system in terms of its Fourier Coefficients[69]. The Widrow's adaptive linear combiner is useful in conducting Fourier analysis.

Researchers worked towards development of NN architectures for control purposes as well the methods to tune the network parameters to ensure the design of stable controller. The principle in NN models are an adaptation of the natural set of neurons, where each neuron predicts an output by weighing up the evidence of truths from fed inputs and shifting the gradient of the resulting function based on an additive bias term, a squashing unit applies a nonlinear transformation to the linearly combined inputs to produce a desired bounded, and constant nonlinear output. By combining a large sum of these simple component connections across the input space and forwarding them through the layers of the network neuron nodes, we obtain a function which approximates the continuous function to an acceptable bounded error, given that there are enough nodes in the network layers. The NN with offline adjusted parameters was commonly used as the nonlinear control law of the process to be controlled. Commonly, the offline estimation techniques for adjusting the parameters of the network using gradient descent optimization methods was presented in the early literature. It was observed that online tuning of parameters introduced high dimensional nonlinearity, even to the linear time invariant (LTI) systems with the linear feedback control. Thus, the control community practicing the formal stability theory considered the application of gradient optimization techniques introduces the instability mechanisms to the process being controlled. Hence, methods based on Lyapunov stability theory was introduced to tune network parameters to obtain stable network architectures which avoids iterative training procedures. The NN based direct adaptive tracking control architecture was proposed for a class of continuous time nonlinear dynamic systems by Slotine et.al [15] .It was suggested that the process of learning and control were never to be carried out simultaneously to ensure proper functioning of dynamics of the system being controlled. The controller design emphasized on tuning the parameters based on the deviations measured from the process to its desired performance. The controller design using nonlinear network structures, which included both NNs and fuzzy logic systems F. L. Lewis [70] is also a major field of study.

Sanner and Slotine's work addresses the direct adaptive tracking control using NNs as controller structures. Additionally, his work highlights the NN architecture for continuous time nonlinear dynamical systems for which an explicit linear parameterization of the uncertainty in the dynamics is either unknown or impossible [15]. The structure proposed is direct adaptive controllers; hence no explicit attempt to determine a model of the process dynamics is made. The controller directly tune its adjustable parameters in response to measured deviations of the process from its desired behavior. Slotine et al. suggested the NN architecture with Gaussian Radial Basis Functions (GRBF) to adaptively compensate for the plant nonlinearities[71][72]. They also suggested gradient descent methods for the adjustment of the parameters but found no systematic way of ensuring when these methods can be successful. It was not recommended to perform learning and control simultaneously unless the system ensures the network training algorithm does not couple destructively with the natural dynamics of the process being controlled. Slotine et al.'s research was to develop stable adaptive architectures capable of exploiting analog network designs for the control of continuous-time nonlinear dynamic systems. The two major aspects of the research were (I) hardware implementation of the NNs for solving real-time control problems (II) treating the problem in the context of adaptive systems theory, avoiding iterative training procedures in favor of provably stable techniques for continuously tuning the network parameters. It is shown that application of Lyapunov stability theory guarantees a given level of performance for the system, but also highlights the relation between performance and the various free parameters in the network. Slotine et al. mainly views NNs only as a method of implementing function approximation strategies in a massively parallel, analog computing paradigm offered by these models, especially if implemented in electronic or optical hardware.

Slotine et al. applies multivariable Fourier analysis and Whittaker/Shannon sampling theory, on networks of GRBF, arranged on a regular lattice on R^n, can uniformly approximate sufficiently smooth functions on closed, bounded subsets. (In the design of Radial Basis Functions(RBF) networks it is common to establish a one to one correspondence between nodes in the hidden layer and the points in the training set). Gaussian rabial basis functions are particularly attractive: they are bounded, strictly positive and absolutely integrable on R^n, and they are their own Fourier transforms (modulo a scale factor). Authors suggested a constructive procedure for selecting the centers and variances of a finite number of Gaussian nodes so that the resulting network is capable of uniformly approximating the required function to a chosen tolerance everywhere on a prespecified subset. It was also highlighted that, as the network consist of only a finite number of nodes, its approximation capabilities can be guaranteed only on a subset of the entire plant state space.

Slotine et.al. presented an adaptive component and nonadaptive component in controller design. The nonadaptive component is designed with a constant gain PD controller and a sliding controller in the algorithm to ensure global stable control strategy. A smooth transition between the adaptive to nonadaptive modes of operation has been designed for the regions of the state space where the network has poor approximating capability. Slotine et al. work provides the scalability to large classes of multi input and input-output linearizable dynamic systems. The work mainly focuses on the approximation problem, highlighting the critical feature as the ability to place prior bounds on the degree of uniform approximation accuracy a tuned network can guarantee.

Sanjay Mistry et al. investigated NN designs with static, dynamic gradient updates for identification and control of time varying and time invariant load cases of a four bar linkage system driven by permanent magnet DC motor (PMDC) through a flexible coupling, described by nonlinear dynamical differential equations [73]. This paper has presented the implementation of K.S. Narendra et al. , a series-parallel NN identifier is modeled for continuous time systems [14]. Mistry et al. provides the explanation for the use of neural identifier can reduce the system sensitivity to sensor degradation and noise through reduced dependence on feedback and increases use of its predicted output. Authors presented a NN identifier and three NN Controller designs. (I) Static Neural Control with the static back propagation technique, with one controller

NN; (II) Single Dynamic Neural Control is designed considering dynamic update scheme representing indirect control using single controller NN; (III) multiple feedforward and feedback networks to incorporate prior information about system structure. The selection of the number and type of inputs is motivated by the ARMA modelling approach for nonlinear systems. The inputs to the identifier are selected by making a discrete linear approximation of the plant and then adding additional past values to accommodate nonlinearities and neglected higher order dynamics. The NN identifier structure used four past values of the actual plant outputs for Time varying case and two previous inputs for time invariant case, one present and one past input to the plant and the present angular position of link. Time invariant load case was simulated with online NN identifier along with Static Controller/ Single dynamic neural controller. The NN structure $\mathfrak{N}^3_{4,20,10,1}$; learning rate 0.001 and sampling frequency 155Hz, finished identification task in 15000secs. Static Controller and Single dynamic neural controller structure $\mathfrak{N}^3_{4,20,10,1}$ were designed with learning rate 0.001 and sampling frequency 155Hz & 75 Hz respectively. Feedback -Feedforward control was not performed. Time variant load case was simulated with online NN identifier along with Static Controller/ Single dynamic neural controller/Feedback-feedforward controller. The NN structure $\mathfrak{N}^3_{7,40,20,1}$; learning rate 0.001 and sampling frequency 80Hz, identification completion time was not specified. Static Controller and Single dynamic neural controller structure $\mathfrak{N}^3_{8,40,20,1}$ were designed with learning rate 0.001 and sampling frequency 77Hz & 12 Hz respectively. Feedback -Feedforward control was performed with 3 identical Feedforward NNs with structure $\mathfrak{N}^3_{1,10,5,1}$ and feedback neural controller structure $\mathfrak{N}^3_{6,10,5,1}$ were designed with learning rate 0.001 and sampling frequency 23Hz. Feedforward and feedback learning controller includes multiple NNs to learn dynamic characteristics of the system including the inertial terms, the centrifugal and Coriolis terms, the gravity terms, the friction terms, and even the unmodeled dynamics. The activation function $f(x) = \frac{(1-x^2)}{2}$ is used. The authors show that the dynamic neural schemes are found to consistently perform much better as compared to the static address that NN identifiers can be trained in advance. Reduced order linear observers are also being investigated for use in dynamic schemes for the time invariant load case for improved convergence and learning features. Authors suggest that a priori knowledge about the system structure can be incorporated into feedforward NN and is suggested to be more adaptable to robotics application. The time invariant load cases are less complex compared to time variant load cases; the sampling frequency is limited for the time varying case for this reason, allowing for only lower velocity set points; The learning time for the time invariant system is found to be approximately 10 times shorter as compared to the time varying case. Time varying case still needs accurate identifier for stability. The authors address the theoretical issues of error convergence and stability, which still needs to be established for NN designs. The topics of research as per the author is generating good partial derivatives, parameter convergence and disturbance rejection.

Kawato et at used hierarchical NN model to control the movements of robot arms. The objective of their study was to determine a desired trajectory in the visual coordinates, and transformation of the same to the joint coordinates and determination of the motor commands [74]. This work provided was an extension of the learning control of the robotic manipulator by an inverse dynamics model developed with three-layer neural network which did not consider any a priori knowledge of the dynamical structure of the controlled object. The improvised hierarchical NN model provided a feedforward control method for large scale complex system. Saad et al. presented studies on trajectory tracking problem to control SCARA robot the nonlinear dynamic model of a robot using neural networks implemented in a DSP based controller[75]. Saad's work provides detailed implementation details two DOF direct drive SCARA experimental setup to test the tracking performances. The discrete dynamics of the plant model is considered. The estimation of the inverse dynamics is performed using NN without any need of parametric model-based dynamics. A simple four-layer MNN is used with each node activated with nonlinear sigmoidal function as similarly applied by K.S. Narendra et al.[14]. The Multilayer recurrent networks are used to estimate the dynamics of the system and the inverse dynamic method. Another Multilayer recurrent NN is used to estimate the joint accelerations. The training state considered large number of input-output measurements around 5000 samples. The training is performed offline using backpropagation using the Brayden-Fletcher-Goldfarb-Swann (BFGS) minimization algorithm. The objective is to minimize the quadratic error using variable step gradient method. Author claims that there is no need to estimate the parameters of the system as in case of direct and indirect adaptive controls as they are computationally expensive to be applied in real time. The control process is applied to a two DOF SCARA robot using DSP controller. The trajectory tracking error are within $\pm 1^0$ and torques expected at manipulator joints were free of chattering. However, the NN structure implementation details were not clearly specified and the performance was not compared with any previous methods to quantify the result. Still the faster NN convergence criteria was not discussed.

As RNN was used extensively for the identification and control applications. The learning algorithms used for RNN are usually based on computing the gradient of a cost function with respect to the weights of the network. One of the learning algorithm, BPTT is a generalization of back propagation for static networks in which one stores the activations of the units while forward propagation. The backward phase is also backward in time and recursively uses these activations to compute the required gradients. Unfortunately, the networks it can deal with have limited storage capabilities for dealing with general sequences, thus limiting their representational power. [76] discussed the long-term storage of definition bits of information into the state variable of the dynamic system referred to as information latching. They suggested the use of hyperbolic attractors (Eg: Hyperbolic tangent functions) (activation functions) to store state information. The paper presents the results for problems of vanishing gradient problem for an RNN (unfolded in time is just a very deep feedforward network with standard weights). The alternate algorithms to gradient descent such as time-weighted pseudo Newton and the discrete error propagation algorithms was suggested by authors.

The paper by Jin et al. [77] proposed a fast neural learning and control of general class of discrete time nonlinear systems. The paper addresses the problem of simultaneous online identification and control in NN based adaptive control systems. The paper uses MNN with feedforward connections. A suitable extension of the concept of input-output linearization of discrete time nonlinear systems is used to develop the control schemes for both output tracking and model reference control purposes. The paper highlights that the main drawback with simultaneous identification and control is the extremely slow convergence of the back-propagation algorithm. Hence, the authors suggest fully decoupled weight learning algorithm, derived based on the extended Kalman filter for MNNs. The multilayered feedforward network the neurons are organized into layers with no feedback or lateral connections. Authors adapted hyperbolic tangent sigmoidal function as activation functions in this paper. The author with the aim of fast convergence of learning introduces a fully decoupled recursive estimation learning algorithm named as weight-decoupled extended Kalman filter (WDEKF). WDEKF shows advantages associated with the computation, storage and it can be integrated into the parallel structure of the network is similar to the conventional back propagation. However, WDEKF is computationally complex than the gradient descent based back propagation algorithm. The authors are yet to provide the theoretical analysis of the convergence of the weight learning algorithm, and the stability of the adaptive control schemes.

Branko proposed a new very fast algorithm for synthesis of a new structure of discrete time NNs and implemented to control RRTR robot structure [78]. Branko suggested that a generalization of NN models can be helpful in the classification and categorization of NN. He mentioned that in the field of identification and control of dynamical systems specifically for robot systems there exist four possibilities in learning a dynamic system such as (I) multilayer feedforward net with pattern learning (FFN-pattern), multilayer feedforward net with batch learning (FFN-batch), feedforward net with external recurrence and pattern learning (RecN-pattern), and feedforward net with external recurrence and batch learning (RecN-batch). The paper presents five main objectives (I) Suggests combinations of feedforward and feedback NN models; (II) Introduction to combination of input, interaction and output activation functions; (III) Use of input and interaction time varying signal distribution results in relatively small number of neurons in NN models with large classification and learning power and applications of time varying input and interaction activation functions; (IV) The fourth objective I to develop time discrete NN model with time delay elements (or memory) in both feedforward and feedback structures; (V) realization of one step learning iteration approach; aiding online learning. The paper contributed in deriving a new algorithm for one step learning in the supervised learning process and a new algorithm for direct inverse modeling in the case where a number of system inputs are different to a number of system outputs.

K. S. Narendra presented a survey on Neural networks for control [79]. His work highlighted the neural structures such as MNN and radial basis function network (RBFN) used for approximating continuous functions using input-output data. Paper provides the features of both the networks. Paper indicates that the massively parallel nature of the MNN permits computation to be performed at high rates; as they can approximate nonlinear maps to any desired degree of accuracy when applied to identification and control problems[80], [81]. The NN's are advantageous as approximation functions compared to conventional methods such as polynomials, trigonometric series, splines and orthogonal functions. NN's architecture is more fault intolerant; less sensitive to noise and easy hardware implementation. Barron's [82] work provided partial theoretical justification for using NN and a measure of boundedness. Barron's view is that as the dimensionality of input space (input vector) increases, it is clear that MNN are preferable to approximation schemes in which the adjustable parameters arise linearly. However, for practical controllers for dynamical system, both MNN and RBFN require substantially fewer parameters for a desired degree of accuracy. The theoretical principles for the design of identifiers especially nonlinear ARMA (NARMA) with two different variants (NARMA L1 & NARMA L2) and controllers using NNs was proposed [83]. He provides the implicit function theorem the properties of the linearized system around the equilibrium state can be extended to the nonlinear domain; allowing for disturbance rejection, decoupling of multivariable systems, and adaptation using multiple models. He states that static backpropagation is adequate for identification, but the adjustment of controller parameters in the feedback loop requires dynamic gradient methods. Multilayer Perceptron provide an input-output representation of the plant in ARMA form for the plants where inputs and outputs are available as network inputs. The spacial aspects of the system is represented using Cerebellar model articulation controller (CAMC), it noted for fast online learning. Both networks provide static mapping between inputs and outputs. Chengyu et al. also suggest the need for dynamic systems to represent the autoregressive aspect of dynamic systems. Authors introduced a method incorporating the analytical knowledge of the plant in a RNN to develop Model Based Recurrent Neural Networks (MBRNN) for modeling nonlinear dynamic systems [84]. Hirasawa et al. presented a new control method of nonlinear systems based on impulse responses of Universal Learning Networks (ULN) [85]; considered as superset of NNs. ULNs consist of a number of interconnected nodes where the nodes may have any continuously differentiable nonlinear functions and each pair of nodes can be connected by multiple branches with arbitrary time delays. Author's prove that dynamics of higher quality such as quick response, quick damping and small steady state error can be achieved easily by the proposed control method.

Adding a new dimension to the study of the identification of the system was the study provided by [86] on the comparison of traditional and NN approaches for Stochastic Nonlinear system identification. Traditional and biologically inspired model structures are compared for their effectiveness to identify the complex stochastic multi input multi output (MIMO) nonlinear system. The two NN models, the state space and input-output model structures were considered for the study. The conventional models considered are Auto-Regressive with eXogenous Input model (ARX) and the Nonlinear ARX (NARX). The NN structures such as the Feedforward Multilayer Perceptron (FMLP) with or without teacher forcing i.e., past observations are used for the teacher forcing FMLP (TFFMLP), and past estimates are used for the recurrent FMLP (RFMLP), in the approximation of function[63]. In addition to FMLP, the Recurrent Multilayer Perceptron (RMLP) model structure allows for feedforward links among the nodes of neighboring layers, and recurrent and crosstalk links within the hidden layers which carry time delayed signals; the observations are provided to the input layer for teacher forcing RMLP (TFRMLP) model; and the estimates are fed back for globally recurrent RMLP (GRRMLP) model structure. The simulations results provided by the paper indicated that that the NARX, FLMP and the RMLP models are good candidate structures for low noise nonlinear system identification, while the FMLP model structures is not as effective as the RMLP and the NARX models. However, for the high noise environment case the RMLP model is the most effective.

Schroder et al. presented intelligent identification, modeling, observation and control of nonlinear plants and to use a priori knowledge of the nonlinear plant about the structure, the relevant order, the parameters and knowledge of the nonlinearities placement [87]. A systematic method to design a stable observer for systems with a known linear part and an unknown nonlinearity. The function approximator, RBFN and the general regression NN (GRNN) are used; another type of multidimensional approximation is fuzzy logic. The paper provided the identification of a static nonlinearity considering the structure and the parameters of the linear section of the plant are known. Lyapunov based adaptation law guaranteed stable learning and parameter convergence. Author's claim that the observer approach presented can be used in practical applications to improve the controller's performance. Hence, making the identification problem possible for all linear parameters and nonlinear characteristics simultaneously in case not all system states are measurable.

A NN based adaptive controller with an observer was proposed for the trajectory tracking of robotic manipulators with unknown dynamic nonlinearities; with an assumption that the robotic manipulator has only joint angle position measurements[88]. Linear observer was used to estimate the robot joint angle velocity. NN's were employed to improve the control performance of the controlled system through approximating the modified robot dynamics function. The simulations were performed on two link manipulators with unknown dynamics. Authors discussed about the MLN based NN based adaptive controller and conventional controller with an observer for robot trajectory tracking. This paper had time varying learning rates to improve the adaptation quality in the initial learning phase. Authors claim that the time varying learning rates will not influence the system stability if appropriate learning rates are chosen. The online tuning laws for the MLN provide good approximation to the modified dynamics functions. The feedforward and feedback through NN were observed to be 1.4 msec when executed on PC-200 using VC 6.0 language. The sampling frequency 200 Hz. The stability and tracking error convergence are proved by Lyapunov approach. Author's present a systematic approach to deal with the trajectory tracking control for a robot with unknown dynamics nonlinearities using an observer.

Srivastava et al. presented the new way of function approximation using wavelets along with NNs to form the wavelet NN (WNN) [89]. Considering the existence of two different WNN architectures: (I) Fixed wavelet bases possessing fixed dilation and translation parameters. With only output layer weights being adjustable. (II) Variable wavelet basis whose dilation and translation parameters and output layer weights are adjustable. As the choice of the wavelet basis to be selected approximately because its selection can be critical to approximation performance. [38] presented the concept of using LSTM NNs for dynamic system identification. He addressed two problems (I) vanishing gradient problem and (II) speed of convergence of the identification error. The first problem is mainly solved by the LSTM structure itself and the second one was conquered by using a convex-based LSTM neural networks structure[59].

Daachi et al. proposed a NN adaptive controller to achieve end-effector tracking of redundant robot manipulators. The unknown model of the system is approximated by a decomposed structure NN. Each NN approximates a separate element of the dynamical model. These approximations were used to derive an adaptive stable control law. The parameter adaptation algorithm was derived from the stability study of the closed loop system using Lyapunov approach with intrinsic properties of robot manipulators. Two control strategies are considered. (I) the aim of the controller is to achieve good tracking of the end-effector regard- less the robot configurations. (II) the controller is improved using augmented space strategy to ensure minimum displacements of the joint positions of the robot [90].

Jeen-Shing et al. presented a fully automated recurrent neural network (FARNN) which is capable of self-structuring its network in a minimal representation with satisfactory performance for unknown dynamic system identification and control [91]. The novel recurrent network, consisting of a fully-connected single-layer NN and a feedback interconnected dynamic network, was developed to describe an unknown dynamic system as a state-space representation. Additionally, a fully automated construction algorithm was devised to construct a minimal state-space representation with the essential dynamics captured from the input–output measurements of the unknown system. The construction algorithm integrates the methods of minimal model determination, parameter initialization and performance optimization into a systematic framework that totally exempt trial-and-error processes on the selections of network sizes and parameters.

Rahmani et al. presented adaptive NN output feedback control for flexible multi-link robots[92]. The approach suitable for highly uncertain systems with arbitrary but bounded dimension. The problem of trajectory tracking is solved through developing a stable inversion for robot dynamics using only joint angles measurement and a linear dynamic compensator to stabilize the tracking error for the nominal system. The authors also introduced a high gain observer to provide an estimate tracking error dynamic. A linear in parameter (LIP) NN approximates and eliminate the effect of the unobservable elastic subsystem in the robot dynamics.

The efforts towards the modifications to standard gradient as an essential element of the optimization problem of the adaptive control theory is studied by many researchers. The seminal accelerated gradient method proposed by Yurii Nesterov, in his work "A method of solving a convex programming problem with convergence rate $o(1/k^2)$". Nesterov's original method, or a variant are the standard methods for training deep neural networks. The research on accelerated methods for the time varying parameter update in both continuous time and discrete time domain. Gaudio et al. presented work on accelerated learning in the presence of time varying features with applications to machine learning and adaptive control [93]. The paper aims to develop algorithms for the machine learning problems when time varying features are present, considering that the gradient descent methods are unstable or weakens the convergence guarantees. They propose a new class of online accelerated algorithms that are inspired by 'high order tuners' used in adaptive control. High order tuners consider the time variation of the features and provide guarantees of stability and convergence.

1.4 Common Criteria of Conventional Adaptive control and learning based controllers

Adaptive control and machine learning have evolved in parallel over the past few decades, with significant similarity in goals, problem statements and tools. Machine learning as a field has focused on static systems that improve through experience. The process of learning is captured in the form of a parameterized model, whose parameters are learned in order to approximate a function. Optimization methods are commonly employed to reduce the function approximation error using any and all available data. The field of adaptive control, on the other hand, has focused on the process of controlling engineering systems in order to accomplish regulation and tracking of critical variables of interest (e.g. speed in automotive systems, position and force in robotics, Mach number and altitude in aerospace systems, frequency and voltage in power systems) in the presence of uncertainties in the underlying system models, changes in the environment, and unforeseen variations in the overall infrastructure. The approach used for accomplishing such regulation and tracking in adaptive control is the learning of underlying parameters through an online estimation algorithm. Stability theory is employed for enabling guarantees for the safe evolution of the critical variables, and convergence of the regulation and tracking errors to zero.

Learning parameters of a model in both machine learning and adaptive control occurs using input-output data. In both cases, the main algorithm used for updating the parameters is based on a gradient descent-like algorithm. Related tools of analysis, convergence, and robustness in both fields have a tremendous amount of similarity. As the scope of problems in both fields increases, the associated complexity and challenges increase as well.

Adaptive control main objective is to carry out problems such as estimation or tracking in the presence of parametric uncertainties. The underlying model that relates inputs, outputs and the unknown parameter is assumed to stem from either the underlying physics or from the data driven approaches. In a control problem the goal is to determine a control input so that the output follows a desired output. The control problem consists of constructing an output tracking error alongside establishing a relationship between the output tracking error and adaptation parameter of the controller. Gradient rule is used to minimize the output error and optimize the adaptation parameter leading to minimize the output error. Gradient based methods to solve for estimates of unknown parameters via back propagation, in what would develop into the foundations of NNs. NNs in control systems has expanded to include stabilizing nonlinear dynamical systems. Design and analysis of stable controllers based on neural networks was taken up by the adaptive control community due to the similarities of gradient-like update laws used in neural networks and adaptive control. The adaptive control community developed a well-established literature for the use of neural networks in nonlinear dynamical systems in the 1990s [54–58].

Machine learning considers the supervised learning problems, with regressors and classifiers. A typical approach taken to perform classification or regression is to choose an output estimator (NN) parameterized with adjustable weights. Like adaptive control problem the weights are adjusted using the output error. The use of NNs in the machine learning community greatly expanded as of recent due to the increase in computing power available and an increase in applications[94][51]. RNN while often similar in structure to nonlinear dynamical systems, have historically been trained in a manner like feed-forward neural networks using BPTT. MNN, RNN and RFBN's have been commonly used in identification and control of static and dynamic nonlinear systems[22], [23], [25], [26], [40], [41] and is listed in Table 1.

Table 1: Neural Network architectures used in commonly used in identification and control of static and dynamic nonlinear systems

Network Architectures	References
Multi-layer networks (static nonlinear maps)	[14], [17], [18], [61], [73], [75], [77], [81], [95]–[98][88]
Recurrent neural networks (nonlinear dynamic feedback systems) (discrete dynamical system)	[14], [57], [66], [67], [84], [91][91]
Radial Basis Function Network (RBFN)	[79][15] [80][92][87], [99][2]

The similarities with the update laws, robustness of the update laws, adaptive gains and step sizes are discussed further. The goal of the adaptive control is to design a rule of adjust parameters/weights in an online continuous manner using knowledge of output measurements and output error such that error tends to zero. The gradient like update laws are commonly used for system which can derive a linear relationship between the output error and adaptation parameter. If a time varying dynamic error models a stability approach rather than gradient based is derived using Lyapunov methods[15]. Most of the learning algorithms with NN architectures uses training algorithms use time varying update law in discrete time which adapts gradient descent method listed in the Table 2 and various activation functions listed in Table 3; the selection of the activation functions is purely based on the speed of convergence of the learning process and the accuracy of the estimation observed.

The adaptive parameter update law must ensure robustness in the presence of bounded disturbances. The parameter update laws are equipped with tunable parameters that scales the parameter update law to achieve robustness. Adaptive control method also employs “dead zone” for the update laws to increase the robustness against disturbances[100]. Similar to conventional adaptive control machine learning / learning methods uses regularization to cope with overfitting by including the constraints on the optimization problem. The training processes is often stopped early to deal with overfitting[101], [102]. Early stopping is often seen to be needed for training NN due to their large number of parameters and can act as regularization.

Table 1: Training algorithms commonly used to train NN

Training algorithms	
BPTT, Static and dynamic back propagation, Backpropagation using the Brayden-Fletcher-Goldfarb-Swann (BFGS) minimization	[6], [13], [14], [57], [58], [62], [103][63][75]
Weight-decoupled extended Kalman filter (WDEKF)	[77]

Table 2: Activation Functions commonly used to train NN

Activation Functions		
Logistic functions	$f(x) = \frac{1}{(1 + e^{(-x)})}$	[17], [18], [57], [78], [82]
Nonlinear Sigmoidal functions	$\sigma(x) = \frac{(1 - e^{(-x)})}{(1 + e^{(-x)})}$ $\sigma(x) = \frac{(1 - e^{(-2x)})}{(1 + e^{(-2x)})}$	[14], [73], [75], [82][98]
Radial Basis Functions	$R_i(u) = exp\left[-\sum_{j=1}^{n} \frac{(u_j - c_{ij})^2}{2\sigma_{ij}^2}\right]$ $u \in \mathbb{R}^n$: input $c_i^T = [c_{i1,} \ \cdots \ c_{in}]$; center of the i[th] receptive field	[15], [71], [72], [79], [80]

	σ_{ij} : width of the receptive field	
General	$f(x) = \frac{(1 - x^2)}{2}$	[73]
Hyperbolic tangent sigmoidal function	$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	[77]
Logistic functions	$f(a) = \frac{1}{(1 + e^{(-1.5a)})}$	[88]

Adaptive control methods employ a compact region apriori for the parameters to be updated such that during the learning process the parameters are not allowed to leave that region. In physical systems there are natural constraints which may aid in the design of that region and for non-physical systems the constraints are often engineered by the algorithm designer. A continuous projection algorithm is commonly employed to provide robustness of the adaptive update law in the presence of unmodeled dynamics [104]. Similarly, learning theory has witnessed project gradient descent methods [105], [106].

There also exists the parameter update laws for the algebraic error model which has the property to alter the gain of the standard update law as function of the time varying regressors in Adaptive control methods. The learning methods also posses the adaptive step size methods due the ability of learning algorithms to handle sparse and small gradients by adjusting the step size as a function of features as they are processed online. A common update law for adaptive step size methods can be similar to both conventional adaptive laws and the learning methods as presented by the following parameterizations (I) projected gradient descent (II) ADAGRAD (III) ADAM. It can be noted that the normalization in these update laws is a function of the gradient, which can be compared to the normalization by the regressor [107]–[109].

The stability and convergence tools in adaptive control is developed around Lyapunov functions and the learning methods online efficiency is analyzed using the notion of regret; where regret is seen to map the sum of the time varying convex costs associated with the time varying parameter, minus the cost of the best static parameter estimate [106].

The plant models used to design the adaptive controllers, are sometimes linearized approximations with a certain amount of modelling errors. The robustness to unforeseen perturbations such as unmodelled dynamics and unseen data can vary the operating point. This adds on the requirements of the design of stabilizing controllers to adapt for parametric uncertainties as well as to be robust for the unmodelled dynamics. In addition, constraints on the state and input may also be present in adaptive control problems. Analysis is said to be difficult when considering such unmodeled dynamics and constraints, resulting in non-global guarantees. Most of the authors have produced the variations in the modification in adaptative control laws to ensure robustness to unmodeled dynamics and constraints as well as adapt for parametric uncertainties. The robustness to modelling errors exists in machine learning in which an estimator is constructed from a finite training data set or real time data generated by the system; along with a finite number if tunable parameters. It is then desired that this estimator produces a low prediction error based on a test data set consisting of not just known data, but unseen data as well. The generalization in learning thus refers to the low loss when applied to new data. In particular it can be seen that in specific cases, generalization pertains to stability, where algorithms that are stable and train in a small amount of time result in a small generalization error[110], [111].

Persistence of excitation (PE) of the system regressor in adaptive control is a condition that has been shown to be necessary and sufficient for parameter convergence[112]. A detailed exposition of system identification and parameter convergence in both deterministic and stochastic cases has been presented by many researchers. Another way to think of the PE condition is that it leads to a perfect test error, since it provides for convergence of the parameter error to zero, and therefore zero output/state error once transients decay to zero. In the absence of PE, standard adaptive control algorithms converge to one of the many local minima in the parameter space. Many machine learning problems consider the parameter estimates can be guaranteed to converge to their true values when stochastic perturbations are present. In this context, significant improvements may be possible by leveraging well known concepts in system identification[101]. For example Sarah.et.al. purposely includes a Gaussian random input into a dynamical system in order to provide for PE by construction [113]. Such stochastic perturbations can guarantee a PE condition only in the limit, when infinite samples can be obtained. In order to address the realistic case of finite samples, approaches in machine learning algorithms for system identification and control have attempted to obtain performance bounds with probability[114]. In many, machine learning methods, including reinforcement learning, there exist explicit modifications to update laws to promote exploration of the parameter (weight) space. These modifications include restarting trajectories with random initial conditions, adding random perturbations to algorithms, and driving the system towards a non-zero error regions. This preference of exploration and learning over stability is motivated by the desire to find optimal parameters of a system.

Given the enormous number of similarities in problem statements, tools, concepts, and algorithms, it is natural to examine what the benefits are that accrue by combining insights obtained in these two different communities; conventional adaptive control and machine learning methods.

Conclusion

As applications become more complex, the processes to be controlled are increasingly characterized by poor models, distributed sensors and actuators, multiple subsystems, high noise levels and complex information patterns. The difficulties encountered in designing controls for such processes can be broadly classified under three headings (1) complexity (2) nonlinearity (3) uncertainty. This research attempted to investigate the evolution of learning based controller design for nonlinear dynamical system specifically a robotic manipulator. Starting with applying identifier NN's as parameterized nonlinear maps; conclusive proofs for MNNs capability of approximating any continuous function on a compact set with precision; and further improvisation of MNNs with dynamics components to be compatible for dynamical systems. The use of NN structures in control problems has entered the mainstream of control theory as a natural extension of adaptive control systems. The recent developments show the use of MNN, RNN and RBFN's in identification and control of static and dynamic nonlinear systems. NN applications in closed loop control are fundamentally different from the open loop applications. In control applications it is necessary to show the stability of the tracking error as well as boundedness of the NN weight estimation errors.

References:

[1] C. Smith and H. I. Christensen, “Robot manipulators,” *IEEE Robot. Autom. Mag.*, vol. 16, no. 4, pp. 75–83, 2009.

[2] Y. P. Pane, S. P. Nageshrao, J. Kober, and R. Babuška, “Reinforcement learning based compensation methods for robot manipulators,” *Eng. Appl. Artif. Intell.*, vol. 78, no. December 2018, pp. 236–247, 2019.

[3] T. Hisa, “Adaptive control of robot manipulators - A review,” in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, 1986, pp. 183–189.

[4] K.J. Astrom, “Interactions between excitation and unmodeled dynamics in adaptive control,” in *The 23rd IEEE Conference on Decision and Control, Las Vegas, Nevada, USA*, 1984, pp. 1276–1281.

[5] A. A. Mullin, “Reviewed Work(s): Principles of Neurodynamics. by Frank Rosenblatt,” *Taylor Fr. Ltd. behalf Math. Assoc. Am.*, vol. 70, no. 5, pp. 433–434, 1963.

[6] B. L. Deekshatulu, “Adaptation and Learning in Automatic Systems,” *IEEE Trans. Comput.*, no. November, 1972.

[7] K. . Narendra, “Adaptation and Learning in Automatic Systems,” *IEEE Trans. Automat. Contr.*, vol. 19, no. 2, pp. 173–174, 1974.

[8] S. Arimoto, S ; Kawamura, S ; Miyazaki, F ; Tamaki, “Learning control theory for dynamical systems,” in *24th IEEE Conference on Decision and Control*, 1985, pp. 1375–1380.

[9] J. J. E. ; W. L. Slotine, “On The Adaptive Control of Robot Manipulators,” *Int. J. Rob. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[10] J. Yuh, “The effect of unmodeled flexible dynamics in discrete time model refernce adaptive control of a flexible single-link robot,” in *26th IEEE Conference on Decision and Control*, 1987, no. December, pp. 1037–1038.

[11] Chae An, C. Atkeson, and J. Hollerbach, “Experimental determination of the effect of feedforward control on trajectory tracking errors,” in *Proceedings. 1986 IEEE International Conference on Robot and Automation, San Francisco, CA, USA*, 1986, pp. 55–60.

[12] P. J. Werbos, “Neural networks for control and system identification,” *Proc. 28th IEEE Conf. Decis. Control*, vol. 1, pp. 260–265, 1989.

[13] P. J. Werbos, "Backpropagation and Neurocontrol: A Review and Proscpects," in *International 1989 Joint Conference on Neural Networks,* 1989, pp. 209–216.

[14] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[15] R. M.Sanner and J. E.Slotine, "Gaussian Networks for Direct Adaptive Control," *IEEE Trans. NEURAL NETWORKS*, vol. 3, no. 6, pp. 837–863, 1992.

[16] E. Rios-Patron and R. D. Braatz, "On the 'Identification and control of dynamical systems using neural networks'.," *IEEE Trans. Neural Netw.*, vol. 8, no. 2, p. 452, 1997.

[17] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Contr. Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[18] A. R. Barron, "Approximation Bounds For Superpositions Of A Sigmoidal Function," *Proceedings. 1991 IEEE Int. Symp. Inf. Theory*, vol. 39, no. 3, pp. 85–85, 1991.

[19] M. Norgaard, O. Ravn, and N. K. Poulsen, "NNSYSID and NNCTRL tools for system identification and control with neural networks," *Comput. Control Eng. J.*, vol. 12, no. 1, pp. 29–36, 2001.

[20] S. Kim, H., Shim, D., & Sastry, "Nonlinear Model Predictive Control for Unmanned Aerial Vehicles," in *American Control Conference*, 2002, vol. Proceeding, no. 5, pp. 3576–3581.

[21] J. Belikov, S. Nõmm, E. Petlenkov, and K. Vassiljeva, "Application of neural networks based SANARX model for identification and control liquid level tank system," *Proc. - 2013 12th Int. Conf. Mach. Learn. Appl. ICMLA 2013*, vol. 1, pp. 246–251, 2013.

[22] H. K. Sahoo, P. K. Dash, and N. P. Rath, "NARX model based nonlinear dynamic system identification using low complexity neural networks and robust H∞ filter," *Appl. Soft Comput. J.*, vol. 13, no. 7, pp. 3324–3334, 2013.

[23] Z. J. Fu, W. F. Xie, X. Han, and W. D. Luo, "Nonlinear systems identification and control via dynamic multitime scales neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 11, pp. 1814–1823, 2013.

[24] O. P. Ogunmolu, X. Gu, S. Jiang, and N. R. Gans, "A real-time, soft robotic patient positioning system for maskless head-and-neck cancer radiotherapy: An initial investigation," *IEEE Int. Conf. Autom. Sci. Eng.*, vol. 2015-Octob, pp. 1539–1545, 2015.

[25] E. Kayacan, E. Kayacan, and M. Ahmadieh Khanesar, "Identification of nonlinear

dynamic systems using type-2 fuzzy neural networks - A novel learning algorithm and a comparative study," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1716–1724, 2015.

[26] M. Chen, G. Tao, and B. Jiang, "Dynamic Surface Control Using Neural Networks for a Class of Uncertain Nonlinear Systems with Input Saturation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 26, no. 9, pp. 2086–2097, 2015.

[27] M. Ben Nasr and M. Chtourou, "Neural network control of nonlinear dynamic systems using hybrid algorithm," *Appl. Soft Comput. J.*, vol. 24, pp. 423–431, 2014.

[28] J. Yu, K. Weng, G. Liang, and G. Xie, "A vision-based robotic grasping system using deep learning for 3D object recognition and pose estimation," *2013 IEEE Int. Conf. Robot. Biomimetics*, no. December, pp. 1175–1180, 2013.

[29] A. Domahidi, M. N. Zeilinger, M. Morari, and C. N. Jones, "Learning a feasible and stabilizing explicit model predictive control law by robust optimization," *IEEE Conf. Decis. Control Eur. Control Conf.*, pp. 513–519, 2011.

[30] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search," in *IEEE International Conference on Robotics and Automation, Stockholm, Sweden*, 2016, pp. 529–535.

[31] N. Wang, J. Sun, M. Han, Z. Zheng, and M. J. Er, "Adaptive Approximation-Based Regulation Control for a Class of Uncertain Nonlinear Systems Without Feedback Linearizability," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3747–3760, 2017.

[32] Z. M. Fadlullah *et al.*, "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *Ieee Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[33] K. Katyal, I. J. Wang, and P. Burlina, "Leveraging Deep Reinforcement Learning for Reaching Robotic Tasks," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 490–491, 2017.

[34] H. Wang, P. X. Liu, S. Li, and D. Wang, "Adaptive neural output-feedback control for a class of nonlower triangular nonlinear systems with unmodeled dynamics," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3658–3668, 2018.

[35] T. Ergen and S. S. Kozat, "Efficient online learning algorithms based on LSTM neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 8, pp. 3772–3783,

2018.

[36] P. Hénaff, V. Scesa, F. Ben Ouezdou, and O. Bruneau, “Real time implementation of CTRNN and BPTT algorithm to learn on-line biped robot balance: Experiments on the standing posture,” *Control Eng. Pract.*, vol. 19, no. 1, pp. 89–99, 2011.

[37] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 1780, pp. 1735–1780, 1997.

[38] Y. Wang, “A new concept using LSTM Neural Networks for dynamic system identification,” *Proc. Am. Control Conf.*, pp. 5324–5329, 2017.

[39] F. Musumeci *et al.*, “An Overview on Application of Machine Learning Techniques in Optical Networks,” *IEEE Commun. Surv. Tutorials*, pp. 1–27, 2018.

[40] H. Yu, J. Peng, and Y. Tang, “Identification of Nonlinear Dynamic Systems Using Hammerstein-Type Neural Network,” *Math. Probl. Eng.*, vol. 2014, pp. 1–9, 2014.

[41] X. Han, W. F. Xie, Z. Fu, and W. Luo, “Nonlinear systems identification using dynamic multi-time scale neural networks,” *Neurocomputing*, vol. 74, no. 17, pp. 3428–3439, 2011.

[42] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, M. A. Bolender, and E. Lavretsky, “Connections Between Adaptive Control and Optimization in Machine Learning,” *arXiv Prepr. arXiv1904.05856*, pp. 1–18, 2019.

[43] H. Durrant-Whyte, “Practical adaptive control of actuated spatial mechanisms,” in *IEEE International Conference on Robotics and Automation*, 1985, pp. 650–655.

[44] K. Y. Lim and M. Eslami, “New Controller Designs for Robot Manipulator Systems,” in *American Control Conference*, 1985, pp. 38–43.

[45] K. S. Narendra and L. S. Valavani, “Direct and Indirect Adaptive Control,” *IFAC Proc. Vol.*, vol. 11, no. 1, pp. 1981–1987, 1978.

[46] P. Van Den Hof, “Closed-Loop Issues in System Identification,” *IFAC Proc. Vol.*, vol. 30, no. 11, pp. 1547–1560, 1997.

[47] S. ARIMOTO and M. TAKEGAKI, “An Adaptive Method for Trajectory Control of manipulators,” in *IFAC Proceedings Volumes*, 1981, vol. 14, no. 2, pp. 1921–1926.

[48] M. R. Rokui and K. Khorasani, “Adaptive tracking control of a flexible link manipulator using a discrete-time nonlinear model,” in *Proceedings Of The American Control Conference*, 1997, no. June, pp. 1848–1853.

[49] M. A. Cohen and S. Grossberg, “Absolute stability of global pattern formation and parallel

memory storage by competitive neural networks," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-13, no. 5, pp. 815–826, 1983.

[50] S. M. Yadav, S. K. Shastri, G. B. Chakravarthi, V. Kumar, A. Divya Rao, and V. K. Agrawal, "A Fast, Parallel Algorithm for Fully Overlapped Allan Variance and Total Variance for Analysis and Modeling of Noise in Inertial Sensors," *IEEE Sensors Lett.*, vol. 2, no. 2, pp. 1–4, 2018.

[51] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018.

[52] B. Recht, "A Tour of Reinforcement Learning: The View from Continuous Control," *Annu. Rev. Control. Robot. Auton. Syst.*, vol. 2, no. 1, p. Control-053018-023825, 2019.

[53] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 997–1007, 1997.

[54] H. Xu and S. Jagannathan, "Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 24, no. 3, pp. 471–484, 2013.

[55] J. Dornheim, N. Link, and P. Gumbsch, "Model-Free Adaptive Optimal Control of Episodic Fixed-Horizon Manufacturing Processes using Reinforcement Learning," *arXiv:1809.06646v3*, pp. 1–12, 2019.

[56] Yi-Jen Wang and Chin-Teng Lin, "Recurrent learning algorithms for designing optimal controllers of continuous systems," *IEEE Trans. Syst. Man, Cybern. - Part A Syst. Humans*, vol. 31, no. 3, pp. 362–372, 2001.

[57] F. J. Pineda, "Generalisation of backpropagation to recurrent neural networks," *Physic Rev. Lett. 18*, vol. 59, no. 19, pp. 2229–2232, 1987.

[58] Paul J. WERBOS, "Backpropagation Through Time : what it does and how it does," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1561, 1990.

[59] S. Sen and A. Raghunathan, "Approximate computing for Long Short Term Memory (LSTM) neural networks," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2266–2276, 2018.

[60] P. Gallman, "An iterative method for the identification of nonlinear systems using a Uryson model," *IEEE Trans. Automat. Contr.*, vol. 20, no. 6, pp. 771–775, 1975.

[61] D. Psaltis, A. Sideris, and A. Yamamura, "A multilayered neural network controller,"

IEEE Control Syst. Mag., vol. 8, no. 2, pp. 17–21, 1988.

[62] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 79, no. 8, pp. 2554–2558, 1982.

[63] K. S. N. and K. Parthasarathy, “Gradient methods for the optimization of dynamical systems containing neural networks,” *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 252–262, 1991.

[64] G. A. Carpenter, M. A. Cohen, S. Grossberg, T. Kohonen, and G. Palm, “Computing with Neural Networks,” *Science (80-.).*, vol. 235, no. 4793, pp. 1226–1228, 1987.

[65] P. Tai and H. A. Ryaaiotaki-boussalis, “The Application of Neural Networks to Control Systems: A Survey,” in *ACSSC*, 1990, pp. 148–151.

[66] B. Kosko, “Bidirectional associative memories,” *IEEE Trans. Syst. Man. Cybern.*, vol. 18, no. 1, pp. 49–60, 1988.

[67] Acevedo-Mosqueda, M. Elena, Yáñez-Márquez, Cornelio, Acevedo-Mosqueda, and M. Antonio, “Bidirectional associative memories: Different approaches.,” *IACM Comput. Surv.*, vol. 45, no. 2, pp. 1–30, 2013.

[68] Kurt Hornik, Maxwell Stinchcombe, and Halbert White, “Multilayer Feedforward Networks are Universal Approximators,” *Neural Networks*, vol. 2, pp. 359–366, 1989.

[69] S. R. Chu, R. Shoureshi, and M. Tenorio, “Neural networks for system identification,” *IEEE Control Syst. Mag.*, vol. 10, no. 3, pp. 31–35, 1990.

[70] F. L. Lewis, “Nonlinear Network Structures for Feedback Control,” *Asian J. Control*, vol. 1, no. 4, pp. 205–228, 2010.

[71] S. Lin, X. Liu, Y. Rong, and Z. Xu, “Almost optimal estimates for approximation and learning by radial basis function networks,” *Mach. Learn.*, vol. 95, no. 2, pp. 147–164, 2014.

[72] T. Poggio and F. Girosi, “Neural Networks for approximation and learning,” *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[73] S. I. Mistry and S. S. Nair, “Identification and Control Experiments Using Neural Designs,” *IEEE Control Syst.*, vol. 14, no. 3, pp. 48–57, 1994.

[74] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, “Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics,” *IEEE Control Syst. Mag.*, vol. 8, no.

2, pp. 8–15, 1988.

[75] M. Saad, P. Bigras, L. A. Dessaint, and K. Al-Haddad, “Adaptive Robot Control Using Neural Networks,” *IEEE Trans. Ind. Electron.*, vol. 41, no. 2, pp. 173–181, 1994.

[76] Y. Bengio, P. Simard, and P. Frasconi, “Learning Long-term Dependencies with Gradient Descent is Difficult,” *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[77] L. Jin, P. N. Nikiforuk, and M. M. Gupta, “Fast Neural Learning and Control of Discrete-Time Nonlinear Systems,” *IEEE Trans. Syst. Man. Cybern.*, vol. 25, no. 3, pp. 478–488, 1995.

[78] B. M. Novakovic, “Discrete Time Neural Network Synthesis Using Input and Output Activation Functions,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 26, no. 4, pp. 533–541, 1996.

[79] K. S. Narendra, “Neural networks for control: Theory and practice,” *Proc. IEEE*, vol. 84, no. 10, pp. 1385–1406, 1996.

[80] S. Chen, C. F. . Cowan, and P. . Grant, “Orthogonal least squares learning algorithm for radial basis function networks,” *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[81] K. Funahashi, “On The Approximate Realization Of Continuous Mappings,” *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.

[82] A. R. Barron, “Approximation and Estimation Bounds for Artificial Neural Networks,” *Mach. Learn.*, vol. 14, no. 1, pp. 115–133, 1994.

[83] K. S. Narendra and S. Mukhopadhyay, “Neural network for system identification,” *IFAC Proc. Vol.*, vol. 30, no. 11, pp. 735–742, 1997.

[84] G. Chengyu and K. Danai, “Model-based recurrent neural network for modeling nonlinear dynamic systems,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 30, no. 2, pp. 344–351, 2000.

[85] K. Hirasawa, J. Hu, J. Murata, and C. Jin, “A new control method of nonlinear systems based on impulse responses of universal learning networks,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 31, no. 3, pp. 362–372, 2001.

[86] K. T. Chong and A. G. Parlos, “Comparison of traditional and neural network approaches to stochastic nonlinear system identification,” *KSME Int. J.*, vol. 11, no. 3, pp. 267–278, 1997.

[87] D. Schroder, C. Hintz, and M. Rau, “Intelligent modeling, observation, and control for nonlinear systems,” *IEEE/ASME Trans. Mechatronics*, vol. 6, no. 2, pp. 122–131, 2001.

[88] Fuchun Sun, Zengqi Sun, and Peng-Yung Woo, “Neural network-based adaptive controller design of robotic manipulators with an observer,” *IEEE Trans. Neural Networks*, vol. 12, no. 1, pp. 54–67, 2002.

[89] S. Srivastava, M. Singh, M. Hanmandlu, and A. N. Jha, “New fuzzy wavelet neural networks for system identification and control,” *Appl. Soft Comput. J.*, vol. 6, no. 1, pp. 1–17, 2005.

[90] B. Daachi and A. Benallegue, “A neural network adaptive controller for end-effector tracking of redundant robot manipulators,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 46, no. 3, pp. 245–262, 2006.

[91] Jeen-Shing Wang and Yen-Ping Chen, “A fully automated recurrent neural network for unknown dynamic system identification and control,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 53, no. 6, pp. 1363–1372, 2006.

[92] B. Rahmani and M. Belkheiri, “Adaptive neural network output feedback control for flexible multi-link robotic manipulators,” *Int. J. Control*, vol. 7179, pp. 1–15, 2018.

[93] J. E. Gaudio, T. E. Gibson, A. M. Annaswamy, and M. A. Bolender, “Accelerated Learning in the Presence of Time Varying Features with Applications to Machine Learning and Adaptive Control,” *arXiv Prepr. arXiv1903.04666*, pp. 1–23, 2019.

[94] S. Sharma, C. Beierle, and S. D’Amico, “Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks,” *IEEE Aerosp. Conf. Proc.*, vol. 2018-March, pp. 1–12, 2018.

[95] A. Rubaai and R. Kotaru, “Adaptation learning control scheme for a high-performance permanent-magnet stepper motor using online random training of neural networks,” *IEEE Trans. Ind. Appl.*, vol. 37, no. 2, pp. 495–502, 2001.

[96] A. Rubaai, R. Kotaru, and M. D. Kankam, “A real-time neural network based controller for brushless DC motor drives,” *IEEE Trans. Ind. Appl.*, vol. 36, no. 2, pp. 475–483, 2000.

[97] G. V. Puskorius, L. A. Feldkamp, and L. I. Davis, “Dynamic neural network methods applied to on-Vehicle idle speed control,” *Proc. IEEE*, vol. 84, no. 10, pp. 1407–1420, 1996.

[98] A. Rubaai and R. Kotaru, “Online Identification And Control of a DC Motor Using

Learning Adaptation of Neural Networks," *IEEE Trans. Ind. Appl.*, vol. 36, no. May-June, pp. 935–942, 2000.

[99] Y. Gao, M. Joo Er, and S. Yang, "Adaptive control of robot manipulators using fuzzy neural networks," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1274–1278, 2001.

[100] K. S. Narendra and A. M. Annaswamy, "A New Adaptive Law for Robust Adaptation Without Persistent Excitation," *IEEE Trans. Automat. Contr.*, vol. 32, no. 2, pp. 134–145, 1987.

[101] J. Sjöberg and L. Ljung, "Overtraining, Regularization, and Searching for Minimum in Neural Networks," *IFAC Proc. Vol.*, vol. 25, no. 14, pp. 73–78, 1992.

[102] J. Sjoberg and L. Ljung, "Overtraining, Regularization, and Searching for Minimum in Neural Networks," 1994.

[103] D. E. Williams, G. E. Rumelhart, and G. E. Hinton, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 9, pp. 533–536, 1986.

[104] G. Kreisselmeier and K. S. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Trans. Automat. Contr.*, vol. 27, no. 6, pp. 1169–1175, 1982.

[105] M. Ravaut and S. Gorti, "Gradient descent revisited via an adaptive online learning rate," *arXiv:1801.09136*, 2018.

[106] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2–3, pp. 169–192, 2007.

[107] S. J. Reddi, S. Kale, and S. Kumar, "On the Convergence of Adam and Beyond," *arXiv:1904.09237*, pp. 1–23, 2018.

[108] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, pp. 1–15, 2014.

[109] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," *arXiv:1212.5701*, 2012.

[110] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," *arXiv:1509.01240*, pp. 1–32, 2015.

[111] B. London, B. Huang, V. Tech, and L. Getoor, "Stability and Generalization in Structured Prediction," *J. Mach. Learn. Res.*, vol. 17, pp. 1–52, 2016.

[112] B. M. Jenkins, A. M. Annaswamy, E. Lavretsky, and T. E. Gibson, "Convergence

Properties of Adaptive Systems and the Definition of Exponential Stability," *arXiv:1511.03222*, 2015.

[113] F. Adiguzel and Y. Yalcin, "Nonlinear Discrete-Time Disturbance Attenuation for Robotic Manipulators with Immersion and Invariance Approach," *ISMSIT 2018 - 2nd Int. Symp. Multidiscip. Stud. Innov. Technol. Proc.*, 2018.

[114] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an Approximate Model Predictive Controller with Guarantees," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 543–548, 2018.