

Review

Open Access



# Evolution of adaptive learning for nonlinear dynamic systems: a systematic survey

Mouhcine Harib<sup>1</sup>, Hicham Chaoui<sup>1</sup>, Suruz Miah<sup>2</sup>

<sup>1</sup>Department of Electronics, Carleton University, Ottawa, ON K1S 5B6, Canada.

<sup>2</sup>Electrical and Computer Engineering, Bradley University, Peoria, IL 61625, USA.

**Correspondence to:** Prof. Hicham Chaoui, Department of Electronics, Carleton University, 7066 Minto Building, Ottawa, ON K1S 5B6, Canada. E-mail: Hicham.Chaoui@carleton.ca

**How to cite this article:** Harib M, Chaoui H, Miah S. Evolution of adaptive learning for nonlinear dynamic systems: a systematic survey. *Intell Robot* 2022;2:37-71. <https://dx.doi.org/10.20517/ir.2021.19>

**Received:** 19 Dec 2021 **First Decision:** 20 Jan 2022 **Revised:** 3 Feb 2022 **Accepted:** 24 Feb 2022 **Published:** 16 Mar 2022

**Academic Editor:** Simon X. Yang **Copy Editor:** Xi-Jun Chen **Production Editor:** Xi-Jun Chen

## Abstract

The extreme nonlinearity of robotic systems renders the control design step harder. The consideration of adaptive control in robotic manipulation started in the 1970s. However, in the presence of bounded disturbances, the limitations of adaptive control rise considerably, which led researchers to exploit some “algorithm modifications”. Unfortunately, these modifications often require a priori knowledge of bounds on the parameters and the perturbations and noise. In the 1990s, the field of Artificial Neural Networks was hugely investigated in general, and for control of dynamical systems in particular. Several types of Neural Networks (NNs) appear to be promising candidates for control system applications. In robotics, it all boils down to making the actuator perform the desired action. While purely control-based robots use the system model to define their input-output relations, Artificial Intelligence (AI)-based robots may or may not use the system model and rather manipulate the robot based on the experience they have with the system while training or possibly enhance it in real-time as well. In this paper, after discussing the drawbacks of adaptive control with bounded disturbances and the proposed modifications to overcome these limitations, we focus on presenting the work that implemented AI in nonlinear dynamical systems and particularly in robotics. We cite some work that targeted the inverted pendulum control problem using NNs. Finally, we emphasize the previous research concerning RL and Deep RL-based control problems and their implementation in robotics manipulation, while highlighting some of their major drawbacks in the field.

**Keywords:** Adaptive control, deep reinforcement learning, manipulators, neural networks, reinforcement learning, robotics



© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.



## 1. INTRODUCTION

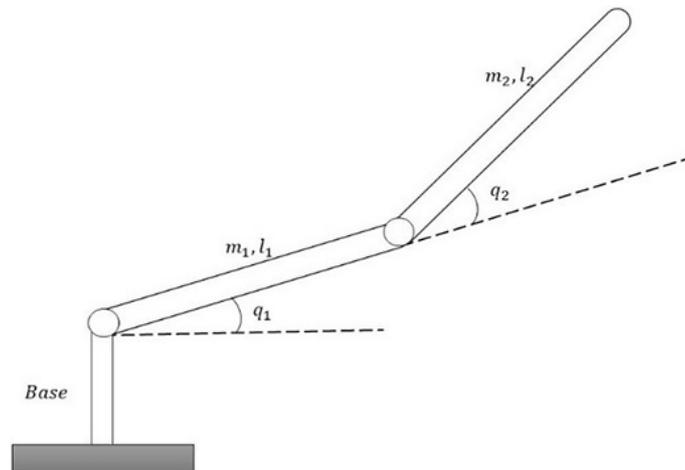
By running a numerical model of a robotic mechanism and its interactions with surroundings, one can define a control algorithm that delivers torque (input) signals to actuators, and that is how a mechanism is able to anticipate the movement. Since robotic systems are extremely nonlinear, the control design is usually a hard step. [Figure 1](#) illustrates a simplified representation of a two-link robot manipulator. Given a system of (dynamic) equation of a robotic system, it contains variables that change when the robot is in motion, which alters the equation mid-task. In this case, a traditional control technique will have to divide the nonlinear mechanism into linear subsystems, which are reasonable for low-speed actions; however, with a high-speed system, their efficacy becomes close to none. For these reasons, adaptive control strategies were first considered.

The system defined by a robot and its controller is complete. Since reconfigurations of the robotic mechanism are needed due to the functional requirements changes, the controller has to adapt to these reconfigurations. In comparison to a non-adaptive control, the adaptive control is able to function without relying on the prior data from the system, since it constantly changes and adjusts to the altered states. That is specifically what makes adaptive control “almost perfect” for systems with unpredictable surroundings, with many probable interferences that could change the system parameters anytime.

In the early years, there were many interests in research and books about the adaptive control<sup>[1-5]</sup> that considered continuous-time systems in most cases. Since 1970, researchers have started dealing with the realization of adaptive control in digital systems. Multiple surveys<sup>[6-8]</sup> show that the consideration of adaptive control systems with discrete-time signals has been around for a while. Many applications of the general adaptive control have been made afterward. There are two fundamental approaches within the adaptive control theory. The first approach is called Learning Model Adaptive Control, where we find the well-known self-tuning adaptive control technique. This approach consists of an improved model of the plant obtained by on-line parameter estimation techniques, and then used in the feedback control. The second approach is called Model Reference Adaptive Control (MRAC). In this case, the controller is adjusted so that the behaviors of the closed-loop system and the preselected model match according to some criterion<sup>[9]</sup>.

Due to the limitations of adaptive control when it comes to bounded disturbances, many researchers turned to “Algorithm Modification” approaches in the 1980s. Typically, these approaches alter least squares adaptation by putting bounds on the error, the parameters, or employing a first order modification of the least squares type of adaptation algorithm. When the observed error is not attributable to an error in the parameter estimations, these strategies effectively turn off or limit the effects of parameter adaptation. The Algorithm Modification techniques essentially perform the same function as the input-output rule-based approaches, but they attempt to have the adaptation algorithm monitor its own level of certainty. The second section of this paper will present more details about the most famous modifications among control researchers, such as Dead-zone modification,  $\sigma$ -modification, and  $\epsilon$ -modification. Unfortunately, these modifications often require a priori knowledge of bounds on the parameters and the perturbations and noise<sup>[10]</sup>. Furthermore, they often improve robustness at the expense of performance.

In a control engineering sense, AI and classical control-based approaches are just different sides of the same coin. Therefore, the limitation of Adaptive control has driven many researchers to consider AI-based controllers. In the 1990s, the field of neural networks was vastly investigated in general, and for control of dynamical systems in particular. The control problem can be formulated as a machine learning (ML) problem, and that is how ML can be mixed with control theory. One of the fundamentally new approaches



**Figure 1.** Simple schematic of a two-link robot manipulator.

is the PILCO approach<sup>[11]</sup>.

Artificial Neural Networks (ANNs) have been explored for a long time in the hopes of obtaining human-like performance in speech and image processing. Several types of Neural Networks (NNs) appear to be promising candidates for control system applications. Multilayer NNs (MLNs), recurrent NNs (RNNs), and the cerebellar model articulation controller (CMAC) are examples of these. The decision of the NN to employ and which training technique to utilize is crucial, and it changes according to the application. The type of NNs most commonly used in control systems is the feedforward MLNs, where no information is fed back during operation. There is, however, feedback information available during training. Typically, supervised learning methods, where the neural network is trained to learn input-output patterns presented to it, are used. Most often, versions of the backpropagation (BP) algorithm are used to adjust the NN's weights during training. The feedforward MLNs are the most often employed NNs in control systems since no information is fed back during operation. During training, however, there is feedback information accessible. In most cases, supervised learning methods are utilized, in which the NN is taught to learn input-output patterns that are provided to it. During training, variants of the BP algorithm are frequently employed to change the NN weights. More details about NNs, for dynamical systems in general and for robotics in particular, are discussed in section 3 of this work.

In robotics, it all boils down to making the actuator perform the desired action. The basics of control systems tell us that the transfer function decides the relationship between the output and the input given the system or plant. While purely control-based robots use the system model to define their input-output relations, AI-based robots may or may not use the system model and rather manipulate the robot based on the experience they have with the system while training or possibly enhance it in real-time as well.

Reinforcement learning (RL) is a type of experience-based learning that may be used in robotics when on-line learning without knowledge of the environment is necessary. The controller may learn which of its possible actions will result in the greatest performance for a particular job for each of its states. If the mobile robot collides with an obstacle, for example, it will learn that this is a poor action, but if it achieves the objective, it will learn that this is a positive activity. Reinforcement or reward is the term for such contextual feedback. The goal of the controller is to maximize its predicted future rewards for state-action pairings,

which are represented by action values. Q-learning is a popular type of RL in which the best policy is learned implicitly as a Q-function. There have been several publications on the use of RL in robotic systems.

This review is organized into 3 sections besides the present introductory chapter and a concluding section. In section 2, we talk about adaptive control limitations for nonlinear systems and introduce the probable drawbacks in the presence of disturbances. We also present the main modifications proposed in the 1980s to overcome these limitations. Section 3 will focus on presenting the work that implemented NNs in nonlinear dynamical systems and particularly in robotics, while we cite some work that targeted the inverted pendulum control problem using NNs. Finally, section 4 emphasizes the previous research concerning RL and Deep RL (DRL) based control problems and their implementation in robotics manipulation, while highlighting some of their major drawbacks in the field.

## 2. ADAPTIVE CONTROL LIMITATIONS - BOUNDED DISTURBANCES

Given a system/plant with an uncertainty set, it is clear that the control objective will be intuitively achievable through either identification, robust control, or a combination of both as in adaptation. The identification is the capability to acquire information in reducing uncertainty. This problem characterization had seen some rigorous analysis over a long period of time and can become very challenging. In 2001, Wang and Zhang<sup>[12]</sup> explored some fundamental limitations of robust and adaptive control by employing a basic first-order linear time-varying system as a vehicle. One can notice that robust control cannot deal with much uncertainty, while the use of adaptive control shows a much better capability of dealing with uncertain parameters and providing better robustness. However, adaptive control requires additional information on parameter evolution and is fundamentally limited to slowly time-varying systems. Furthermore, adaptation is not capable of achieving proximity to the nominal performance when under near-zero variation rates.

### 2.1. Problem statement

The design of adaptive control laws is always under the assumption that system dynamics are exactly specified by models. Hence, when the true plant dynamics is not perfectly described by any model, as expected from a practice point of view, one can only question the real behavior of the control. The robust stability, required for any adaptive control to achieve practical applicability of the algorithms, can be provided when only the modeling error is sufficiently “small”. Unfortunately, stability alone cannot guarantee robustness, since the modeling error appears as a disturbance and usually causes divergence of the adaptive process.

While one of the fundamental fields of application of adaptive control is in systems with unknown time-varying parameters, the algorithms have been proved robust, in the presence of noise and bounded disturbances, only for systems with constant parameters<sup>[13]</sup>. Ideally, when there are no disturbances or noise and when parameters are constant, adaptation shows smooth convergence and stability properties. On the other hand, the adaptive laws are not robust in the presence of bounded disturbances, noise and time-varying parameters.

In order to mathematically state the problem of non-robustness of adaptive control to bounded disturbances, let us start by considering a MIMO system in the form<sup>[14,15]</sup>,

$$\begin{aligned} \dot{x} &= A_{ref}x + B\Lambda \left( u + K^T\Psi(x) \right) + B_{ref}y_{cmd} + \zeta(t) \\ y &= C_{ref}x \end{aligned} \quad (1)$$

where  $\zeta(t) \in \mathbb{R}^n$  is a bounded time-dependant disturbance,  $x \in \mathbb{R}^n$  is the extended system state vector,  $y \in \mathbb{R}^n$  is the controlled system output,  $u \in \mathbb{R}^m$  is the control input and  $\Psi \in \mathbb{R}^N$  is the known N-dimensional regressor vector. We assume  $(A_{ref}, B, B_{ref}, C_{ref})$  are known and  $A_{ref}$  is Hurwitz.  $y_{cmd} \in \mathbb{R}^m$  in this case is a bounded command for  $y$ .  $\Lambda \in \mathbb{R}^{m \times m}$  is a diagonal positive definite matrix and  $K \in \mathbb{R}^{N \times m}$  is a constant matrix, where both matrices represent the matched uncertainties of the system. In addition, we assume that,

$$\|\zeta(t)\| \leq \zeta_{max} \tag{2}$$

and that the disturbance upper bound  $\zeta_{max} \geq 0$  is known and constant.

The control goal is bounded tracking of the reference model dynamics,

$$\begin{aligned} \dot{x} &= A_{ref}x_{ref} + B_{ref}y_{cmd} \\ y &= C_{ref}x_{ref} \end{aligned} \tag{3}$$

driven by a bounded time-dependant command  $y_{cmd} \in \mathbb{R}^m$ .

Based on Equation (1), the control input is selected as,

$$u = -\hat{K}^T \Psi(x) \tag{4}$$

where  $K \in \mathbb{R}^{N \times m}$  is the matrix of adaptive parameters. If we substitute Equation (4) into Equation (1), we get,

$$\dot{x} = A_{ref}x - B\Lambda\Delta K^T \Psi(x) + B_{ref}y_{cmd} + \zeta(t) \tag{5}$$

where,

$$\Delta K = \hat{K} - K \tag{6}$$

is the matrix of estimation errors. The tracking error is  $e = x - x_{ref}$ . Subtracting the reference model dynamics in Equation (3) from that of Equation (1) yields the tracking error dynamics,

$$\dot{e} = A_{ref}e - B\Lambda\Delta K^T \Psi(x) + \zeta(t) \tag{7}$$

The Lyapunov function candidate is selected,

$$V(e, \Delta K) = e^T P e + trace(\Delta K^T \Gamma_K^{-1} \Delta K \Lambda) \tag{8}$$

where  $\Gamma_K = \Gamma_K^T > 0$  represents constant rates of adaptation, and  $P = P^T > 0$  is the unique symmetric positive definite solution of the algebraic Lyapunov equation,

$$A_{ref}^T P + P A_{ref} = -Q \quad (9)$$

with  $Q = Q^T > 0$ . The time derivative of  $V$ , along the trajectories of Equation (7),

$$\dot{V} = -e^T Q e - 2e^T P B \Lambda \Delta K^T \Psi(x) + 2\text{trace}(\Delta K^T \Gamma_K^{-1} \hat{K} \Lambda) + 2e^T P \zeta(t) \quad (10)$$

Applying the trace identity,

$$a^T b = \text{trace}(b a^T) \quad (11)$$

yields,

$$\dot{V} = -e^T Q e + 2\text{trace}(\Delta K^T \{\Gamma_K^{-1} \hat{K} - \Psi e^T P B\} \Lambda) + 2e^T P \zeta(t) \quad (12)$$

Using the following adaptive law yields,

$$\dot{\hat{K}} = \Gamma_K \Psi e^T P B \quad (13)$$

then,

$$\dot{V} = -e^T Q e + 2e^T P \zeta(t) \leq -\lambda_{\min}(Q) \|e\|^2 + 2\|e\| \xi_{\max} \lambda_{\max}(P) \quad (14)$$

and, consequently,  $V < 0$  outside of the set,

$$E_0 = \left\{ (e, \Delta K) \mid \|e\| \leq \frac{2\xi_{\max} \lambda_{\max}(P)}{\lambda_{\min}(Q)} = e_0 \right\} \quad (15)$$

Hence, trajectories  $[e(t), \Delta K(t)]$ , of the error dynamics in Equation (7) coupled with the adaptive law in Equation (13), enter the set  $E_0$  in finite time and stay there for all future times. However, the set  $E_0$  is not compact in the  $(e, \Delta K)$  space. Moreover, it is unbounded since  $\Delta K$  is not restricted. Inside the set  $E_0$ ,  $V$  can become positive and consequently, the parameter errors can grow unbounded, even though the tracking error norm remains less than  $e_0$  at all times. This phenomenon is caused by the disturbance term  $\zeta(t)$ . It shows that the adaptive law in Equation (13) is not robust to bounded disturbances, no matter how small the latter is.

In the 1980s, several studies analyzed the stability of adaptive control systems and many of them concentrated on linear disturbance-free systems<sup>[16-20]</sup>. The results, however, are not completely satisfactory, since they do not consider the cases where disturbances are present, which could completely change the

efficiency of the control system, even when very small, leading to instability. In the years that followed, there have been many attempts to overcome the limitations of adaptive control in the presence of bounded disturbances. In these published papers<sup>[21-25]</sup>, it is shown that unmodelled dynamics or even very small bounded disturbances can cause instability in most of the adaptive control algorithms.

Many efforts to design robust adaptive controllers in the case of unknown parameters have consistently progressed along two different shapes<sup>[26-31]</sup>. In the first, the adaptive law is altered so that the overall system has bounded solutions in the presence of bounded disturbances. The second relies on the persistent excitation of certain relevant signals in the adaptive loop. The next subsections will present some of the main “modifications” proposed to enforce robustness with bounded disturbances.

## 2.2. Dead-zone modification

In many physical devices, the output is zero until the magnitude of the input exceeds a certain value. Such an input-output relation is called a dead-zone<sup>[32]</sup>. In a first approach to prevent instability of the adaptive process in the presence of bounded external disturbances, Egardt<sup>[26]</sup> introduced a modification of the law so that adaptation takes place when the identification error exceeds a certain threshold. The term dead-zone was first proposed by Narendra and Peterson<sup>[18]</sup> in 1980, where the adaptation process stops when the norm of the state error vector becomes smaller than a prescribed value. In 1980, the study was initiated by Narendra to determine an adaptive control law that ensures the boundedness of all signals in the presence of bounded disturbances, in the case of continuous systems. In the study of Peterson and Narendra<sup>[30]</sup>, they highlight the cruciality of the proper choice of the dead zone for establishing global stability in the presence of an external disturbance. A larger dead zone implies that adaptation will take place in shorter periods of time, which also means larger parameter errors and larger output. One of the assumptions made in this paper is that a bound of the disturbance can be determined even though the plant parameters are unknown. The adaptive law shall consider that the module of the augmented error is not greater than the bound plus an arbitrary positive constant. Hence, the only knowledge needed to calculate the size of the dead zone is the bound of the disturbance, which can be computed<sup>[30]</sup>. It is also worth noting that no prior knowledge of the zeros of the plant’s transfer function is needed to find the bound.

Samson<sup>[28]</sup> presented a brief study in 1983 based on all his previous works and the analysis of Egardt in his book. Although his paper was only concerned with the stability analysis and not the convergence of the adaptive control to an optimal state, he was able to efficiently introduce a new attempt to use the possible statistical properties of the bounded disturbances. The three properties  $P_1$ - $P_3$  should be verified by the identification algorithm and are similar to the ones demanded for the disturbance-free cases, but less restrictive. The first property states that the identified vector has to be uniformly bounded, which prevents the system from diverging faster than exponentially. The second property ensures that the prediction error remains relatively small, which indicates that the “adaptive observer” transfer function is very similar to that of the system. Finally, the third property allows the control of the time-varying adaptive observer of the system.

In 1983, a modified dead-zone technique was proposed in Bunich’s research<sup>[33]</sup> and was widely used. This modification permits a size reduction of the residual set for the error, hence, simplifying the convergence proof. The drawback is the necessary, yet restrictive, knowledge of a bound on the disturbance in order to appropriately determine the size of the dead-zone.

The work of Peterson and Narendra invigorated a new study by Sastry<sup>[34]</sup> where he examined the robustness aspects of MRAC. Sastry used the same approach to show that a suitably chosen dead-zone can also stabilize

the adaptive system against the effects of unmodelled dynamics. Though, the error between the plant and the model output does not converge to zero but rather to a magnitude less than the size of the dead-zone. In other terms, no adaptation takes place when the system is unable to distinguish between the error signal and the disturbance.

The issue in the dead-zone modification is that it is not Lipschitz, which may cause high-frequency oscillations and other undesirable effects when the tracking error is at or near the dead-zone boundary. In 1986, Slotine and Coetsee<sup>[35]</sup> proposed a “smoother” version of the dead-zone modification. Unfortunately, we were not able to get a hold of a copy of this paper, but the major idea was explained in his book in 1990<sup>[32]</sup>.

### 2.3. $\sigma$ -modification

The dead-zone modification assumes a priori knowledge of an upper bound for the system disturbance. On the other hand, the  $\sigma$ -modification scheme does not require any prior information about bounds for the disturbances. This modification was proposed by Ioannou and Kokotovic<sup>[36]</sup> in 1983, which Ioannou referred to later as “fixed  $\sigma$ -modification”. The modification basically adds damping to the ideal adaptive law. They introduced the modification by adding a decay term  $-\sigma I\theta$  to the disturbance-free integral adaptive law, where  $\sigma$  is a positive scalar to be chosen by the designer. The stability properties with the modification were established based on the existence of a positive constant  $p$  such that, for  $\sigma > p$ , the solutions for the error and adaptive law equations are bounded for any bounded initial condition. A conservative value of  $\sigma$  has to be chosen in order to guarantee  $\sigma > p$ . It was also shown that the modification yields the local stability of an MRAC scheme when the plant is a linear system of relative degree one and has unmodeled parasitics.

However, even though the robustness achievement is done smoothly and in a simpler way with the  $\sigma$ -modification scheme, there is the potential destruction of some of the convergence properties, since there is no more asymptotic convergence and the fine tracking error is confined within a bounded region. Consequently, many additional modifications have been suggested later, motivated by the aforementioned drawback of the  $\sigma$ -modification. In 1986, Ioannou and Tsakalis<sup>[37]</sup> proposed the “switching  $\sigma$ -modification”. In contrast to Ioannou’s earlier work<sup>[38,39]</sup>, the switching of  $\sigma$  from 0 to  $\sigma_0$  is modified so that  $\sigma$  is a continuous function of  $|\theta(t)|$  [Equation (16)], since the previous modification choices forced the adaptive law to be discontinuous, which might not guarantee the existence of a solution and would probably cause oscillations on the switching surface during implementation. Hence, the continuous switching, as shown in [Figure 2](#), replaces the discontinuous one and is defined in Equation (16),

$$\sigma_s = \begin{cases} 0 & \text{if } |\theta(t)| < M_0 \\ \sigma_0 \left( \frac{|\theta(t)|}{M_0} - 1 \right) & \text{if } M_0 \leq |\theta(t)| \leq 2M_0 \\ \sigma_0 & \text{if } |\theta(t)| > 2M_0 \end{cases} \quad (16)$$

where  $M_0 > 0$ ,  $\sigma_0 > 0$  are design constants and  $M_0$  is chosen to be large enough so that  $M_0 > |\theta^*|$ .

In 1992, Tsakalis<sup>[40]</sup> employed the  $\sigma$ -modification to target the adaptive control problem of a linear, time-varying SISO plant. The signal boundedness for adaptive laws was guaranteed using the  $\sigma$ -modification, normalization and a sufficient condition. The condition relates the speed and the range of the plant parameter variations with the  $\sigma$  value and simplifies the selection of the design parameters in the adaptive law.

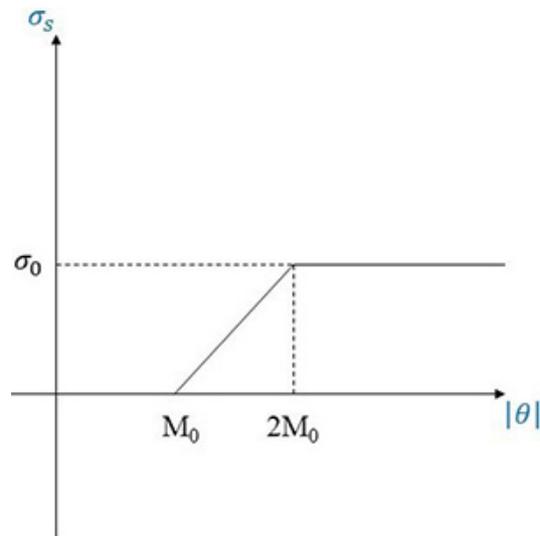


Figure 2. Switching  $\sigma$ -modification (continuous).

In a more recent study, He *et al.*<sup>[41]</sup> opted to revisit the fundamental  $\sigma$ -modification scheme and propose a qualitative analysis for all the scenarios where this modification can lead to perfect tracking, and where it can allow proper modification of the adaptive laws. The analysis method pre-supposes the existence of a Lyapunov function for an extended system, as shown in the reference<sup>[42]</sup>. The efficacy of the proposed analysis was demonstrated in a Robust adaptive control system in order to detect its global asymptotic convergence under the fixed  $\sigma$ -modification scheme. When it comes to simulation results, the system shows asymptotic convergence of its trajectories without the modification; however, it may lose its asymptotic stability when the feedback gain and the modification gain are not well designed when using the modification. The recovery of the global asymptotic convergence is primarily dependant on the proper design of both gains, as shown in their last simulation.

#### 2.4. $\epsilon$ -modification

The downside of the  $\sigma$ -modification is that when the tracking error becomes small enough, the adaptive parameters have an inclination to revert to the origin, which undoes the gain values that caused the tracking error to become small in the first place. In order to overcome this undesirable effect, Narendra and Annaswamy<sup>[43]</sup> developed the  $\epsilon$ -modification. The suggested modification was motivated by that given in the work of Ioannou and Kokotovic<sup>[36]</sup>, which similarly guarantees bounded solutions in the presence of bounded disturbances when the reference input is not persistently exciting, and needs less prior information regarding plant and disturbance. However, the catching point comes when the reference input is persistently exciting and has a sufficiently large amplitude. In this case, as we mentioned earlier, the origin of the error equations is exponentially stable, unlike that in Ioannou's  $\sigma$ -modification. The new adaptive law replaces the  $\sigma$  with a term proportional to the magnitude of the output error, called  $\epsilon$  (or  $e_t$  in the work of Narendra and Annaswamy<sup>[43]</sup>).

Ideally, let's consider the first order plant described with Equation (17),

$$\dot{x}_p = a_p x_p + u \quad (17)$$

where  $a_p$  is an unknown constant. The reference model is defined in Equation (18),

$$\dot{x}_m = -a_m x_m + r \quad (18)$$

where  $a_m > 0$  and  $r$  is a bounded piecewise continuous function. Equation (19) shows how the aim of the adaptive control is to choose the control input,  $u$ , such that the output of the plant approaches that of the model,

$$u(t) = r(t) + \theta(t)x_p(t) \quad (19)$$

where  $\theta$  is the control parameter. Therefore, we deduct the error equations in Equation (20),

$$\begin{aligned} \dot{\epsilon} &= -a_m \epsilon + \varphi x_p \\ \varphi &\triangleq \theta(t) - \theta^* \\ \theta^* &\triangleq -a_m - a_p \\ \epsilon &\triangleq x_p - x_m \end{aligned} \quad (20)$$

and hence, based on Equation (21), the proposed adaptive law can be defined,

$$\dot{\varphi} = -\epsilon x_p - \gamma |\epsilon| \theta \quad \gamma > 0 \quad (21)$$

where  $\epsilon$  is playing a double role since it attempts to decrease the magnitude of the output error while keeping the parameter  $\theta$  or the parameter error  $\varphi$  bounded. The choice of the Lyapunov function, in Equation (22), gives the time derivative of  $V$ ,

$$\dot{V} = -a_m \epsilon^2 - \gamma |\epsilon| \varphi \theta \quad (22)$$

If we define a set  $D$ ,

$$D \triangleq \left\{ (\epsilon, \varphi) \mid \varphi^2 + \varphi \theta^* + \frac{a_m}{\gamma} |\epsilon| \leq 0 \right\}$$

we then can deduct that  $V \leq 0$  inside the set  $D$ . The modification, which is synthesized by the additional term  $-|\epsilon|\theta$  in the adaptive law, shows that the set  $D$  is compact, which allows us to apply LaSalle's theorem<sup>[44]</sup> and prove that all solutions of the error equations are bounded.

If we distinguish between the three possible cases based on the reference input states: null, constant or persistently excited, and as mentioned earlier, the third case's application highlights the difference between the proposed modification and the aforementioned  $\sigma$ -modification. When the  $\sigma$ -modification is used to adjust the control parameter  $\theta$ , in the presence of the disturbance  $v$ , we can set the error equations as in Equation (23),

$$\begin{cases} \dot{\epsilon} = -a_m \epsilon^2 + x_p \phi + v \\ \dot{\phi} = -(x_p \epsilon + \sigma \phi + \sigma \theta^*) \end{cases} \quad (23)$$

where it has been shown that three stable equilibrium states exist in case  $x_m = 0$ , none of them is the origin, and has a single equilibrium state whose distance from the origin decreases as the amplitude of  $x_m$  increases, and that is in case  $x_m$  is a constant. Which clearly highlights the addition of the  $\epsilon$ -modification.

## 2.5. Summary

Basically, the approaches discussed above reduce the effects of parameter adaptation when the measure error is not due to an error in the parameter estimates. They contribute to either parameter error, noise error, high-frequency unmodelled dynamics error, or disturbances, which consist of anything undescribed by the three previous groups<sup>[45]</sup>. A brief comparison of all the aforementioned modification techniques is shown in Table 1.

Considering the robustness problem, one can see that the disturbance is generated internally, which makes it dependable in the actual plant's input-output signals. Particularly, the disturbance will grow unboundedly if the adaptive system is unstable and the input-output signals are growing without bound. Videlicet, the stability problem becomes internal and signal dependant. Thus, the boundedness of the disturbance should not be presumed, which proves that, despite the intrinsic results shown in the previous literature, the aforementioned approaches do not necessarily solve the robustness problem in the presence of bounded disturbances<sup>[46]</sup>.

Over the years, the adaptive controllers have proven themselves effective, especially in the process that can be modeled linearly with slowly time-varying parameters relative to the system's dynamics. The 1980s were the peak of theoretical research on this case. On the other hand, many practical examples can be found in these research<sup>[8,47-52]</sup>.

An overview of some practical examples of adaptive control applications in two different fields, thermal exchange and robotics, is given in Table 2<sup>[53-57]</sup>. We would also like to refer the readers to a very concise survey written by Åström<sup>[8]</sup> in 1983 for more practical examples of the applications of adaptive control. In addition, adaptive controllers are extremely practical and fruitful when it comes to servo systems that have large disturbances, like load changes, or uncertainties, like frictions, and that have measurable states. The number one practical field in that era was robotics<sup>[53-57]</sup>.

Obviously, adaptive controllers are not the "perfect" solution to all control problems. For instance, they do not provide stability for systems where parameter dynamics are at least the same magnitude as the system's dynamics. The controller robustness can be improved by employing artificial intelligence (AI) techniques, such as fuzzy logic and neural networks<sup>[58-60]</sup>. Essentially, these methods approximate a nonlinear function and provide a good representation of the nonlinear unknown plant<sup>[61]</sup>, although it is typically used as a model-free controller. The plant is treated as a "black box", with input and output data gathered and trained

**Table 1. Stability analysis of each modification technique**

<b>Dead-zone modification</b>	<b><math>\sigma</math>-modification</b>	<b><math>\epsilon</math>-modification</b>
<ul style="list-style-type: none"> <li>Developed based on adaptation hibernation principle.</li> <li>Stops adaptation when the error touches the boundary of a compact set <math>\beta_d</math>: <math>\beta_d = \{(e, \Delta K^T), e \in R^n, \Delta K \in R^{N \times m} \mid \ e\  \leq e_d\}</math></li> <li>Adaptation will be disabled once reaches <math>e_d</math></li> <li>Stability is guaranteed outside of <math>\beta_d</math></li> <li>The adaptive law is defined in both conditions as:</li> </ul> <p><u>Drawbacks:</u> a prior knowledge about the upper bound of the disturbance is required</p>	<ul style="list-style-type: none"> <li>Adding a damping term to the adaptation law: <math>K = \Gamma_k^{-1}(\Psi e^T PB - \sigma K)</math>, where <math>\sigma &gt; 0</math></li> <li>Takes different forms depending on the choice of sigma</li> <li>The Lyapunov function derivative is negative under some conditions that define a compact set <math>\beta_\sigma</math>: <math>\beta_\sigma = \{(e, \Delta K^T), e \in R^n, \Delta K \in R^{N \times m} \mid \ e\  \leq e_\sigma \wedge (\ \Delta K\ _F \leq \Delta K_\sigma)\}</math></li> <li>Error UUB is guaranteed and boundedness of all adaptive gains is also guaranteed</li> </ul> <p><u>Drawbacks:</u> the damping term addition may not be convenient in some situations</p>	<ul style="list-style-type: none"> <li>Adding an error dependent leakage term to the law: <math>K = -\Gamma_k e^T BP(\Psi - \epsilon K)</math>, where <math>\epsilon &gt; 0</math></li> <li>Reduces the unbounded behavior of the adaptive law</li> <li>Following the same argument as in sigma modification: the Lyapunov function derivative is negative under certain conditions that define a compact set <math>\beta_\epsilon</math>: <math>\beta_\epsilon = \{(e, \Delta K^T), e \in R^n, \Delta K \in R^{N \times m} \mid \ e\  \leq e_\epsilon \wedge (\ \Delta K\ _F \leq \Delta K_\epsilon)\}</math></li> <li>Error UUB is guaranteed and boundedness of all adaptive gains is also guaranteed</li> <li>The upper bound of the set is determined by the upper bound of the disturbance</li> </ul>

**Table 2. Practical examples of adaptive control implementation**

<b>Approach</b>	<b>Employed by...</b>
<b>Robotic manipulators</b>	
MRAC	Dubowsky and DesForges <sup>[53]</sup> (1979) and Nicosia and Tomei <sup>[55]</sup> (1984)
STAC	Koivo and Guo <sup>[56]</sup> (1983)
Adaptive algorithm	Dubowsky <sup>[54]</sup> (1981) and Horowitz and Tomizuka <sup>[57]</sup> (1986)
<b>Other applications</b>	
MRAC	Harrell et al. <sup>[49]</sup> (1987) and Davidson <sup>[47]</sup> (2021)
STAC	Davison et al. <sup>[48]</sup> (1980) and Harris and Billings <sup>[52]</sup> (1981)
Direct AC	Zhang and Tomizuka <sup>[50]</sup> (1985)
Function Blocks	Lukas and Kaya <sup>[51]</sup> (1983)

MRACL Model Reference Adaptive Control; STAC: self-tuning adaptive control.

on. The AI framework addresses the plant's model after the training phase, and can handle the plant with practically no need for a mathematical model. It is feasible to build the complete algorithm using AI techniques, or to merge the analytical and AI approaches such that some functions are done analytically and the remainder are performed using AI techniques<sup>[62]</sup>.

### 3. NEURAL NETWORKS FOR DYNAMIC SYSTEMS

The sophisticated adaptive control techniques that have been created complement computer technology and offer significant potential in the field of applications where systems must be regulated in the face of uncertainty. In the 1980s, there was explosive growth in pure and applied research related to NN. As a result, MLN and RNN have emerged as key components that have shown to be exceptionally effective in pattern recognition and optimization challenges<sup>[63-68]</sup>. These networks may be thought of as components that can be employed efficiently in complicated nonlinear systems from a system-theoretic standpoint.

The topic of regulating an unknown nonlinear dynamical system has been approached from a variety of perspectives, including direct and indirect adaptive control structures, as well as multiple NN models. Because NN may arbitrarily simulate static and dynamic, highly nonlinear systems, the unknown system is

replaced by a NN model with a known structure but a number of unknown parameters and a modeling error component. With regard to the network nonlinearities, the unknown parameters may appear both linearly and nonlinearly, changing the original issue into a nonlinear robust adaptive control problem.

### 3.1. Neural network and the control of dynamic nonlinear systems

The characteristic of neural networks is that they are quite parallel. They can speed up computations and assist in the solving of issues that need much processing. Since NNs have nonlinear representations and can respond to changes in the environment, they easily reflect physical conditions like industrial processes and their control, whereas precise mathematical models are harder to construct.

One of the few theoretical frameworks for employing NNs for the controllability and stability of dynamical systems has been established by Levin and Narendra<sup>[69]</sup>. Their research is limited to feedforward MLNs with dynamic BP and nonlinear systems with full state information access. Figure 3 presents the proposed architecture of the NNs. Equation (24) considers a system at a discrete-time index  $k$ ,

$$x(k+1) = f(x(k), u(k)) \quad (24)$$

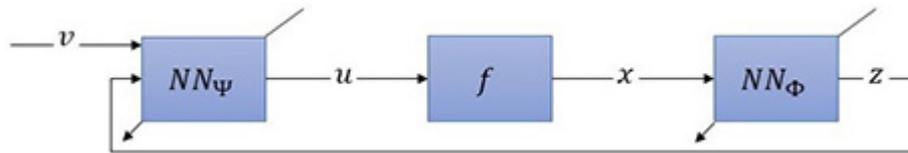
where  $x(k) \in \chi \subset \mathbb{R}^n$ ,  $u(k) \in U \subset \mathbb{R}^r$  and  $f(0,0) = 0$  so that  $x = 0$  is an equilibrium. Conditions are given, in Equation (25), under which the two following NNs can be trained to feedback linearize and stabilize the system.

$$\begin{cases} u = NN_{\Psi}(v, z) \\ z = NN_{\Phi}(x) \end{cases} \quad (z, v) \in \mathbb{R}^n \times \mathbb{R}^r \quad (25)$$

The results are extended to non-feedback linearizable systems. If the controllability matrix around the origin has a full rank, a methodology and conditions for training a single NN to directly stabilize the system around the origin have been devised. Narendra and Parthasarathy<sup>[70]</sup> use NNs to create various identification and controller structures. Although the MLNs represent static nonlinear maps and the RNNs represent nonlinear dynamic feedback systems, they suggest that the feedforward MLNs and RNNs are comparable. They describe four network models of varying complexity for identifying and controlling nonlinear dynamical systems using basic examples.

Sontag proposed an article where he tried to explore the capabilities and the ultimate limitations of alternative NN architectures<sup>[71]</sup>. He suggests that NNs with two hidden layers may be used to stabilize nonlinear systems in general. Intuitively, the conclusion contradicts NNs approximation theories, which claim that single hidden layer NNs are universal approximators. Sontag's solutions are based on the description of the control issue as an inverse kinematics problem rather than an approximation problem.

In 1990, Barto<sup>[72]</sup> drew an interesting parallel between connectionist learning approaches and those investigated in the well-established field of classical adaptive control. When utilizing NNs to address a parameter estimate problem, the representations are frequently chosen based on how nervous systems represent information. In contrast, in a traditional method, issue representation options are made based on the physics of the problem. As opposed to conventional methods, a connectionist approach is dependent on the structure of the network and the correlation between the connectionist weights. A traditional controller may readily include a priori information; however, in NNs, it is often an input-output connection. In both



**Figure 3.** Architecture of the proposed NNs in the work of Levin and Narendra<sup>[69]</sup>. NNs: Neural Networks.

techniques, performance may be assessed using cost functions such as least mean squared error. All of the training data is available at the same time with off-line approaches. However, with on-line approaches, the required feature is continuous learning, and as a result, the methods must be extremely efficient in order to keep up with the changing events over time.

Adaptive NNs have recently been used by a growing number of academics and researchers to construct acceptable control rules for nonlinear systems. An overview of the primarymost recent literature that implemented adaptive NNs-based techniques is discussed in Table 3<sup>[73-82]</sup>.

### 3.2. Inverted pendulum

Many researchers have studied learning control using the inverted pendulum problem. The canonical underactuated system, called the cart-pole system, is illustrated in Figure 4. Because deriving the dynamics is relatively simple, it is considered a basic control issue, yet it still hides some underlying complexity owing to its underactuated character. The multiple obstacles that must be addressed to properly regulate such extremely complex nonlinear unstable systems include severe nonlinearities, variable operating circumstances, structured and unstructured dynamical uncertainties, and external disturbances. The purpose of the control is to balance the pole by moving the cart, which has a restricted range of movements. We distinguish between the position of the cart  $h$  and its velocity  $\dot{h}$ , and the angle of the pole  $\theta$  with its angular velocity  $\dot{\theta}$ .

In 1983, Barto *et al.*<sup>[83]</sup> showed how a system consisting of two neuronlike adaptive elements, associative search element (ASE) and adaptive critic element (ACE), can solve a difficult learning control problem such as the cart-pole system. Their work was based on the addition of a single ACE to the ASE developed by Michie and Chambers in the works of Michie and Chambers<sup>[84,85]</sup>. They have partitioned the state space into 162 boxes. Their simulations revealed that the ASE/ACE system outperformed the boxes system in terms of run time. The system was more likely to solve the problem before it had 100 failures, but the boxes system was less likely to do so. The ASE/ACE system's high performance was nearly completely owing to the ACE's provision of reinforcement throughout the trials. Learning occurs only upon failure with box systems and ASEs without an ACE, which happens less frequently as learning progresses. An ASE can get input on each time step with the ACE in place. The system attempts to access some areas of the state space and avoids others as a result of the learning achieved by this input.

Anderson<sup>[86]</sup> built on the work of Barto *et al.*<sup>[83]</sup> by using a variant of the common error BP algorithm to two-layered networks that learn to balance the pendulum given the inverted pendulum's real state variables as input. Two years later<sup>[87]</sup>, he summarized both aforementioned works by discussing the neural network structures and learning methods from a functional viewpoint and by presenting the experimental results. He described NN learning techniques, which use two functions to learn how to construct action sequences. The first is an action function, which converts the current state into control actions. The second is an evaluation function, which converts the present state into an assessment of that state. There were two sorts of networks that emerged: "action and evaluation" networks. This is an adaptive critic architecture version

**Table 3. Different adaptive NN-based controls in the recent years**

Research	Method/approach	Solved problem
1. Nonaffine nonlinear systems		
Dai <i>et al.</i> <sup>[73]</sup>	Obtaining the implicit desired control input (IDCI), and use of NNs to approximate it	Learning from adaptive NN-based control for a class of nonaffine nonlinear systems in uncertain dynamic environments
Chen <i>et al.</i> <sup>[74]</sup>	The unknown functions are approximated by using the property of the fuzzy-neural control	Adaptive fuzzy-NN (FNN) for a class of nonlinear stochastic systems with unknown functions and a nonaffine pure-feedback form
2. Tracking control		
Dai <i>et al.</i> <sup>[75]</sup>	Radial basis function NNs (RBF-NNs) to learn the unknown dynamics, and adaptive neural control to guarantee the ultimate boundedness (UB)	Stabilization of the tracking control problem of a marine surface vessel with unknown dynamics
Li <i>et al.</i> <sup>[76]</sup>	NNs to approximate the unknown functions, and Barrier Lyapunov function (BLF) for nonstrict-feedback stochastic nonlinear system	Adaptive tracking control for a category of SISO stochastic nonlinear systems with dead zone and output constraint
Cheng <i>et al.</i> <sup>[77]</sup>	Use of NN-based inversion-free controller, and construction of dynamic model using feedforward MLNs	Displacement tracking control of piezo-electric actuators (PEAs)
Ren <i>et al.</i> <sup>[78]</sup>	Use of adaptive neural control, and inclusion of $\sigma$ -modification to the adaptation law to establish stability	Tracking control problem of unknown nonlinear systems in pure-feedback form with the generalized P-I hysteresis input
3. Unknown model/direction		
Luo <i>et al.</i> <sup>[79]</sup>	Implementing three NNs to approximate the value function, control and disturbance policies, respectively	Date-driven $H_\infty$ control for nonlinear distributed parameter systems with a completely unknown model
Liu <i>et al.</i> <sup>[80]</sup>	Two types of BLFs are used to design the controller and analyze the stability	Stabilize a class of nonlinear systems with the full state constraints and the unknown control direction
4. Backstepping design		
Li <i>et al.</i> <sup>[81]</sup>	Adaptive backstepping control and RBF-NNs.	Overcoming the robustness issues of backstepping design and its uncertainty.
5. Discrete-time systems		
Zhang <i>et al.</i> <sup>[82]</sup>	Iterative adaptive dynamic programming algorithm, with two NNs to approximate the costate function and the corresponding control law	Solving the optimal control problem for discrete-time systems with control constraints

NNs: Neural Networks.

In 1991, Lin and Kim integrated the CMAC into the self-learning control scheme that was based on the work of Lin and Kim<sup>[88]</sup>. The CMAC model was originally proposed by Albus<sup>[89-92]</sup> and it was based on models of human memory and neuromuscular control. The CMAC-based technique in the work of Lin and Kim<sup>[88]</sup> is tested using the inverted pendulum problem, and the results are compared to those of Barto *et al.*<sup>[83]</sup> and Anderson<sup>[87]</sup>. The technique has the highest learning speed due to its capability of generalization and good learning behavior. Furthermore, the memory size can be reduced compared to the box-based system. A summarized timeline of the above literature, where NN-based control was implemented to balance the inverted pendulum, is presented in [Figure 5](#).

Many control laws for inverted pendulums have been presented in those research work<sup>[93-95]</sup>, including classical, robust, and adaptive control laws, but they all take structured parametric uncertainty into account. In 2009, Chaoui *et al.*<sup>[96]</sup> proposed an ANN based adaptive control strategy for inverted pendulums that accomplishes asymptotic motion tracking and posture control with unknown dynamics. Two neural networks ANN<sub>x</sub> and ANN <sub>$\theta$</sub>  are designed to control the motion along the  $x$  axis and the pendulum posture with unknown dynamics. [Figure 6](#) shows the block diagram of the proposed system.

Three experiments are carried out to evaluate the performance of the proposed controller. The velocity and posture of the pendulum progressively decrease to zero in the first experiment. The proposed adaptive

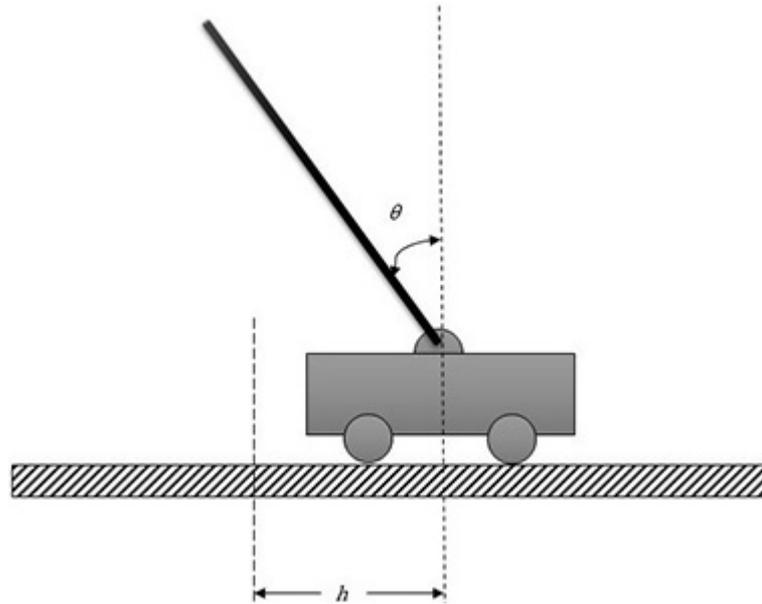


Figure 4. The Cart-pole scheme.

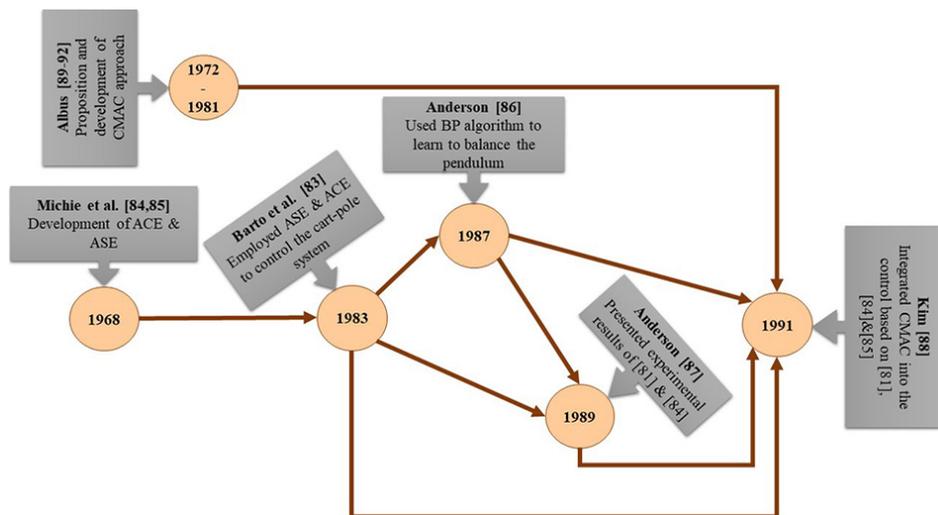


Figure 5. Timeline scheme of the works that kickstarted the use of NNs to control the inverted pendulum. NNs: Neural Networks.

control, on the other hand, produces a smooth control signal. The controllers also deal with friction nonlinearities and accomplish quick error convergence and tracking. The second experiment introduces a starting posture position to test the controller's capacity to correct for a non-zero position error. Posture control takes precedence over motion tracking, as posture is critical for such systems. The purpose of the third experiment is to demonstrate the modularity of the proposed controller in terms of adjusting for external disturbances. The suggested controller's design does not clearly model the induced external disturbance, which generally has a considerable impact on the positioning system's accuracy and generates unacceptably high-frequency oscillations. The controller is able to deal with the unexpected force change

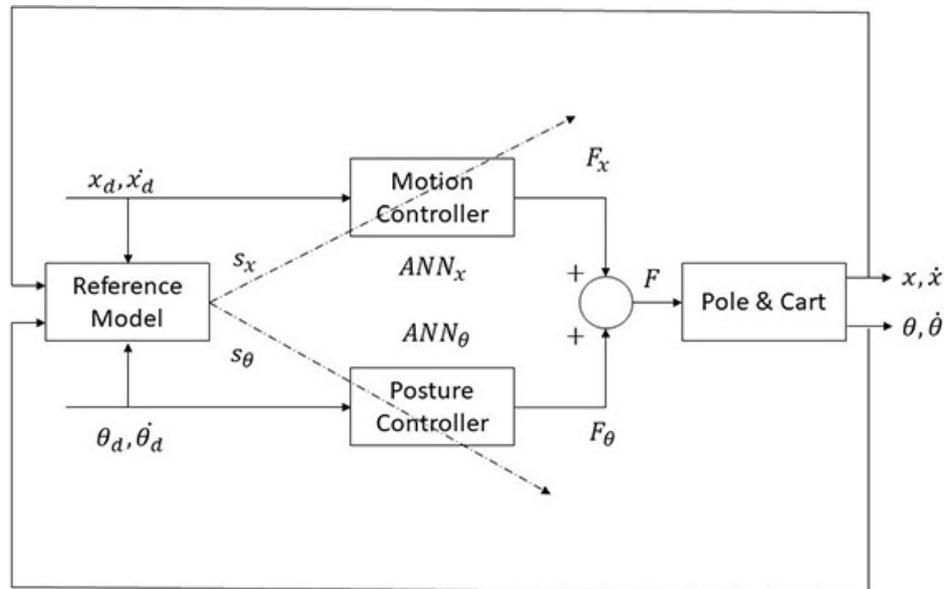


Figure 6. Block diagram of the ANN-based adaptive control scheme<sup>[96]</sup>. ANN: Artificial Neural Network.

successfully. Furthermore, the motion and posture errors are kept to a minimum, resulting in a smooth control signal.

### 3.3. Applications for robotic manipulators

There has been great interest in universal controllers that mimic the functions of human processes to learn about the systems they are controlling on-line so that performance improves automatically. NN-based controllers are derived for robot manipulators in a variety of applications, including position control, force control, link flexibility stabilization and the management of high-frequency joint and motor dynamics. The nature of joint torques must be determined for the end effector to follow the required trajectory as quickly and accurately as feasible, which is a common difficulty for robot manipulators. Both parametric and structural uncertainties necessitate adaptive control. Parametric uncertainties originate from a lack of accurate information about the manipulator's mass characteristics, unknown loads, and load location uncertainty, among other things. Structural uncertainties are a result of the presence of high-frequency unmodeled dynamics, resonant modes, and other structural reservations.

The late 1980s and early 1990s were booming years for both NNs and robotic manipulators research. In this era, the literature survey concerning the application of NNs in robotic manipulators is very rich. Thus, we direct the readers to some interesting approaches in these studies<sup>[97-102]</sup> and the references therein.

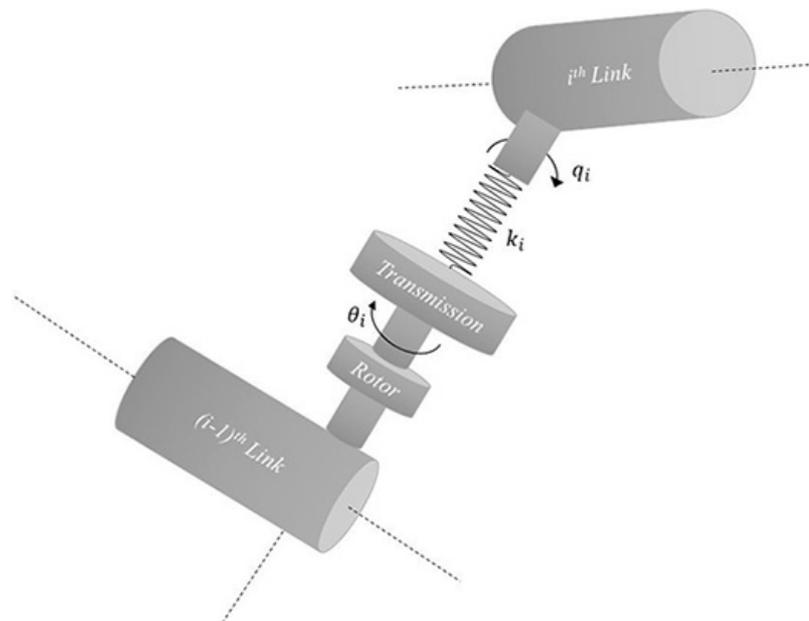
From 1987 to 1989, Miller *et al.*<sup>[103-107]</sup> discuss a broad CMAC learning technique and its application to robotic manipulators' dynamic control. The dynamics do not need to be known in this application. Through input and output measurements, the control scheme learns about the process. The findings show that when compared to fixed-gain controllers, the CMAC learning control performs better. Also, because measured and estimated values must be transformed to discrete form, each variable's resolution and range must be carefully selected, and the number of memory regions handled by each input state in the CMAC architecture is the most important design parameter. In another popular approach, Miller *et al.*<sup>[108]</sup> used CMAC in the real-time control of an industrial robot and other applications. In their network, they utilize hundreds of thousands of adjustable weights that, in their experience, converge in a few iterations.

Huan *et al.*<sup>[109]</sup> examine the issue of building robot hand controllers that are device-dependent. Their argument for a controller like this is that it would isolate low-level control issues from high-level capabilities. They employ a BP algorithm with a single hidden layer comprised of four neurons to achieve this goal. The inputs are determined by the object's size, while the outputs are determined by the grab modes. In this way, they have demonstrated how to build a p-g table using simulation. Another BP architecture was used by Wang and Yeh<sup>[110]</sup> to control a robot model which simulates PUMA560. A network to simulate the plant and a controller network make up their self-adaptive neural controller (SANC). The plant model is trained either off-line with mathematical model outputs or on-line with plant outputs through excitations. The control network is modified by working in series with the plant network during the "controlling and adapting" phase. The control network is also trained off-line in a "memorizing phase" with data from the adapting phase in a random way, which is another element of this training. This trait, according to the authors, aids in overcoming the temporal instability that is inherent with BP. Their numerical findings show that the SANC technique produces good trajectory-tracking accuracy.

Up to the early 2000s, the main goal of robotic manipulators designs was to minimize vibration and achieve good position accuracy, which led to maximizing stiffness. This high stiffness is achieved by using heavy material and a bulky design. As a result, it is demonstrated that heavy rigid manipulators are wasteful in terms of power consumption and operational speed. It is necessary to reduce the weight of the arms and increase their speed of action in order to boost industrial output. As a result of their light weight, low cost, bigger work volume, improved mobility, higher operational speed, power economy, and a wider range of applications, flexible-joint manipulators have gotten much attention. [Figure 7](#) shows a representation of a flexible joint manipulator model.

Controlling such systems, however, still challenges significant nonlinearities, such as coupling caused by the manipulator's flexibility, changing operating conditions, structured and unstructured dynamical uncertainties, and external disturbances. Complex dynamics regulate flexible-joint manipulators<sup>[111-114]</sup>. This emphasizes the need to examine alternate control techniques for these types of manipulator systems in order to meet their increasingly stringent design criteria. Many control laws for flexible joints have been presented in those studies<sup>[115-118]</sup> to solely address (structured) parametric uncertainties. The proposed controllers need a complete a priori knowledge of the system dynamics. Several adaptive control systems<sup>[119-121]</sup> have been proposed to alleviate this necessity. The majority of these control strategies use singular perturbation theory to extend adaptive control theory established for rigid bodies to flexible ones<sup>[122-125]</sup>.

Based on all the above reasons, computational intelligence techniques, such as ANNs and fuzzy logic controllers, have been credited in a variety of applications as powerful controllers of the types of systems that may be subjected to structured and unstructured uncertainties<sup>[126,127]</sup>. As a result, there have been advancements in the field of intelligent control<sup>[128,129]</sup>. Various neural network models have been used to operate flexible-joint manipulators, and the results have been adequate<sup>[130]</sup>. Chaoui *et al.*<sup>[131,132]</sup> developed a control strategy inspired by sliding mode control that uses a feedforward-NN to learn the system dynamics. Hui *et al.*<sup>[133]</sup> proposed a time-delay neuro-fuzzy network. The joint velocity signals were estimated using a linear observer in this system, which avoided the need to measure them directly. Subudhi and Morris<sup>[134]</sup> proposed a hybrid architecture that included a NN for controlling the slow dynamic subsystem and an  $H_\infty$  for controlling the rapid dynamic subsystem. Despite its effectiveness, NN-based control systems are still unable to incorporate any humanlike experience already obtained about the dynamics of the system in question, which is regarded as one of the soft computing approaches' primary flaws.



**Figure 7.** Flexible-joint manipulator model.

Chaoui *et al.*<sup>[135]</sup> suggested an ANN-based control technique in 2009, which used ANNs' learning and approximation skills to estimate the system dynamics. The MRAC is made up of feedforward ( $ANN_{FF}$ ) and feedback ( $ANN_{FBK}$ ) NN-based adaptive controllers. The reference model is built in the same manner as a sliding hyperplane in variable structure control, and its output, which may be regarded as a filtered error signal, is utilized as an error signal to adjust the  $ANN_{FBK}$ 's weights. It comprises a first-order model that specifies the required dynamics of the error between the desired and real load locations, as well as between the motor and load velocity, in order to maintain internal stability. The  $ANN_{FF}$  offers an approximate inverse model for the positioning system, while the  $ANN_{FBK}$  corrects residual errors, assuring the manipulator's internal stability and rapid controller response.

The feedback's learning rate is dependent on the load inertia, which is a flaw in this construction. To improve the stability region of the NN-based controllers, a supervisor is proposed to modify the learning rate of the ANNs. The supervisor also increases the adaptation process's convergence qualities.

Nowadays, the subject of multiple-arms manipulation highlights some interesting progress in using intelligent control approaches. Hou *et al.*<sup>[136]</sup> used a dual NN to solve a multicriteria optimization problem for coordinated manipulation. Li *et al.*<sup>[137,138]</sup> are representatives who operate on several mobile manipulators with communication delays. Some promising approaches, such as LMI and fuzzy-NN controls, were used in both articles<sup>[137,138]</sup>, to improve motion/force performances, which were crucial in multilateral teleoperation applications.

In 2017, He *et al.*<sup>[139]</sup> proposed an Adaptive NN-based controller for a robotic manipulator with time-varying output constraints. The adaptive NNs were utilized to adjust for the robotic manipulator system's uncertain dynamics. The disturbance-observer (DO) is designed to compensate for the influence of an unknown disturbance, and asymmetric barrier Lyapunov Functions (BLFs) are used in the control design process to avoid violating time-varying output constraints. The effects of system uncertainties are successfully corrected, and the system's resilience is increased using the adaptive NN-based controller. The NN estimating errors are coupled with the unknown disturbance from people and the environment to form a combined disturbance that is then approximated by a DO.

In a recent interesting paper, He *et al.*<sup>[140]</sup> attempted to control the vibrations of a flexible robotic manipulator in the presence of input dead-zone. The lumped technique is used to discretize the flexible link system<sup>[141,142]</sup>. A weightless linear angular spring and a concentrated point mass are used to partition the flexible link into a finite number of spring-mass parts. They design NN controllers with complete state feedback and output feedback based on the constructed model. All state variables must be known to provide state feedback. An observer is presented to approximate the unknown system state variables in the case of control with output feedback. In summary, an overview of the evolution of NNs implementation in robotic manipulation is shown in Table 4. Each of these papers has been categorized based on the nature of its approach.

### 3.4. From machine learning to deep learning

ML has transformed various disciplines in the previous several decades, starting in the 1950s. NN is a subfield of ML, a subset of AI, and it is this subfield that gave birth to Deep Learning (DL). There are three types of DL approaches: supervised, semi-supervised, and unsupervised. There is also a category of learning strategy known as RL or DRL, which is commonly considered in the context of semi-supervised or unsupervised learning approaches. Figure 8 shows the classification of all the aforementioned categories.

The common-sense principle behind RL is that if an action is followed by a satisfying state of affairs, or an improvement in the state of affairs, the inclination to produce that action is enhanced, or in other words reinforced. Figure 9 presents a common diagram model of general RL. The origin of RL is well rooted in computer science, though similar methods such as adaptive dynamic programming and neuro-dynamic programming (NDP)<sup>[143]</sup> were developed in parallel by researchers and many others from the field of optimal control. NDP was nothing but reliance on both concepts of Dynamic-Programming and NN. For the 1990's AI community, NDP was called RL. This is what makes RL one of the major NN approaches to learning control<sup>[60]</sup>.

On the other hand, deep models may be thought of as deep-structured ANNs. ANNs were first proposed in 1947 by Pitts and McCulloch<sup>[144]</sup>. Many major milestones in perceptrons, BP algorithm, Rectified Linear Unit, Max-pooling, dropout, batch normalization, and other areas of study were achieved in the years that followed. DL's current success is due to all of these ongoing algorithmic advancements, as well as the appearance of large-scale training data and the rapid development of high-performance parallel computing platforms, such as Graphics Processing Units<sup>[145]</sup>. Figure 10 shows the main types of DL architectures. In 2016, Liu *et al.*<sup>[146]</sup> proposed a detailed survey about DL architectures. Four main deep learning architectures, which are restricted Boltzmann machines (RBMs), deep belief networks (DBNs), autoencoder (AE), and convolutional neural networks (CNNs), are reviewed.

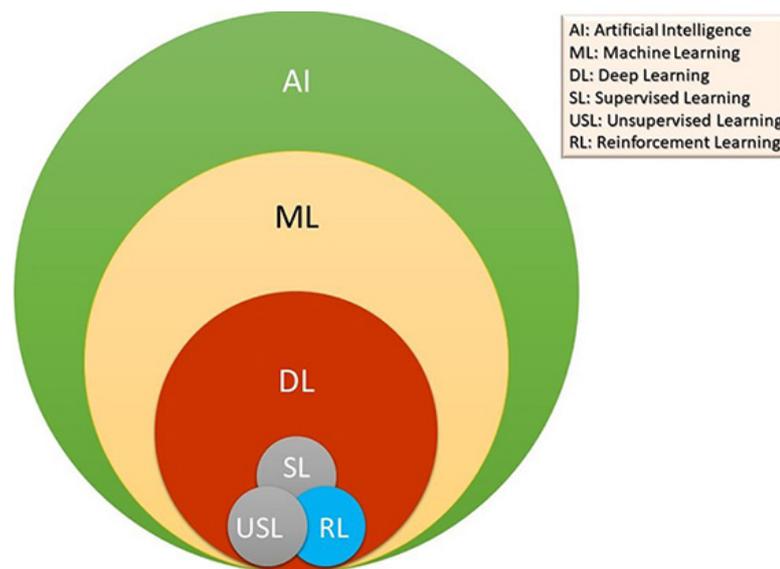
## 4. RL/DRL FOR THE CONTROL OF ROBOT MANIPULATION

DRL combines ANN with an RL-based framework to assist software agents in learning how to achieve their

**Table 4. NN-based control in robotic manipulation - an overview**

Approach	Employed by...
Backpropagation	Elsley <sup>[98]</sup> (1988), Huan et al. <sup>[109]</sup> (1988), Karakasoglu and Sundareshan <sup>[100]</sup> (1990) and Wang and Yeh <sup>[110]</sup> (1990)
CMAC learning	Miller et al. <sup>[103-108]</sup> (1987-1990)
Adaptive NNs/PG table	Huan et al. <sup>[109]</sup> (1988) and He et al. <sup>[139]</sup> (2017)
NNs for flexible joints	Hui et al. <sup>[133]</sup> (2002), Gueaieb et al. <sup>[128]</sup> (2003), Chaoui et al. <sup>[131,132]</sup> (2004), Subudhi and Morris <sup>[134]</sup> (2006), Chaoui et al. <sup>[130]</sup> (2006), Chaoui and Gueaieb <sup>[126]</sup> (2008), He et al. <sup>[140]</sup> (2017) and Sun et al. <sup>[142]</sup> (2017)
NNs for multiple arms	Hou et al. <sup>[136]</sup> (2010), Li and Su <sup>[137]</sup> (2013) and Li et al. <sup>[138]</sup> (2014)
Feedforward and feedback	Chaoui et al. <sup>[135]</sup> (2009)
<b>RNNs</b>	
Hopfield net	Xu et al. <sup>[101]</sup> (1990)
Comparison	Wilhelmsen and Cotter <sup>[102]</sup> (1990)

NNs: Neural Networks; CMAC: cerebellar model articulation controller; RNNs: recurrent NNs.

**Figure 8.** Classification of AI categories.

objectives. It combines function approximation and goal optimization to map states and actions to the rewards they result in. The combination of NN with RL algorithms led to the creation of astounding breakthroughs like Deepmind's AlphaGo, an algorithm that beat the world champions of the Go board game<sup>[147]</sup>.

As mentioned earlier, RL is a powerful technique for achieving optimal control in robotic systems. Traditional optimal control has the drawback of requiring complete understanding of the system's dynamics. Furthermore, because the design is often done offline, it is unable to deal with the changing dynamics of a system during operation, such as service robots that must execute a variety of duties in an unstructured and dynamic environment. The first chapter of this paper has shown that adaptive control, on



Figure 9. Universal model of RL. RL: Reinforcement learning.

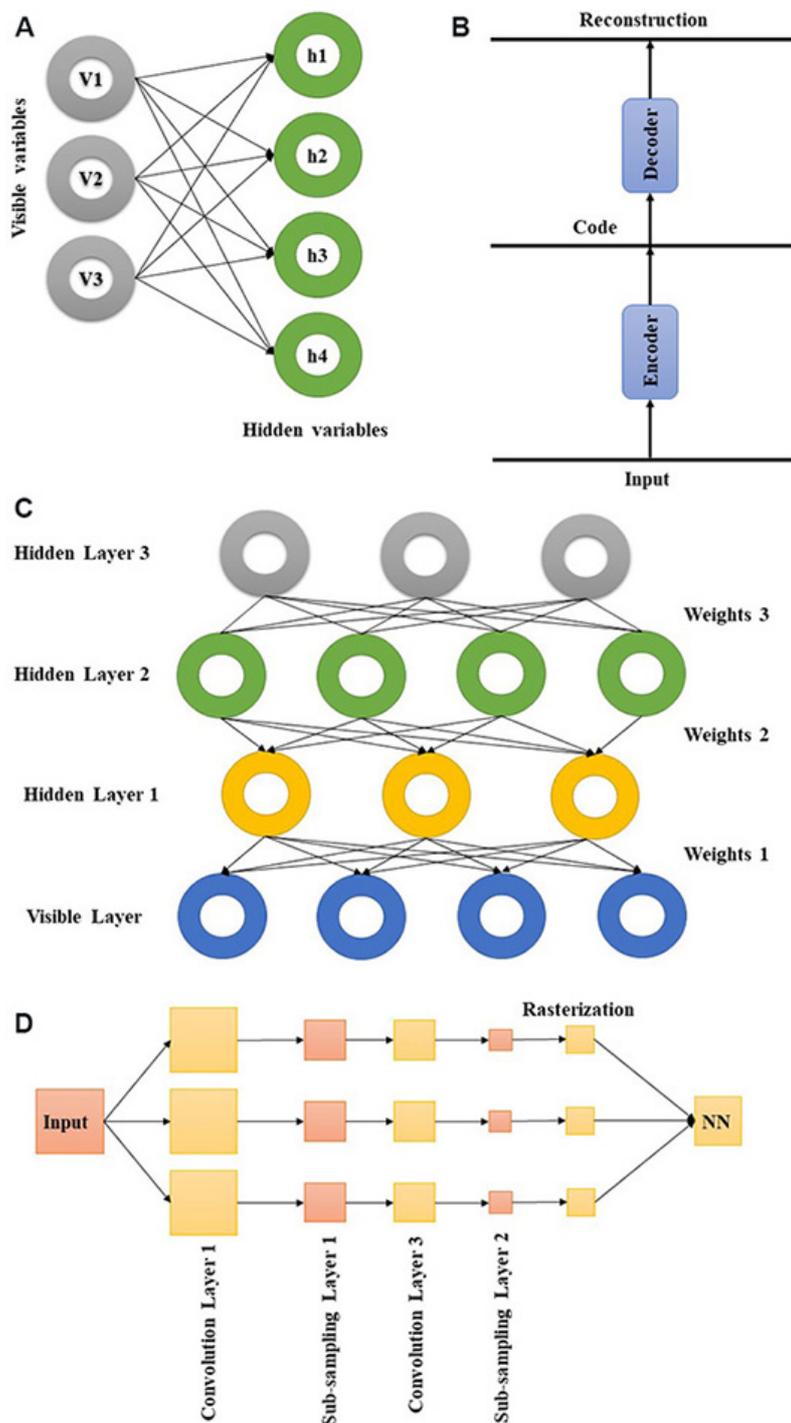
the other hand, is well known for online system identification and control. Adaptive control, on the other hand, is not necessarily optimal and may not be appropriate for applications such as humanoid robots/service robots, where optimality is essential. Furthermore, robots that will be employed in a human setting must be able to learn over time and create the best biomechanical and robotics solutions possible while coping with changing dynamics. Optimality in robotics might be defined as the use of the least amount of energy or the application of the least amount of force to the environment during physical contact. Aspects of safety, such as joint or actuator restrictions, can also be included in the cost function.

#### 4.1. Reinforcement learning for robotic control

The reinforcement learning (RL) domain of robotics differs significantly from the majority of well-studied RL benchmark issues. In robotics, assuming that the true state is totally visible and noise-free is typically impractical. The learning system will have no way of knowing which state it is in, and even very dissimilar states may appear to be quite similar. As a result, RL in robots is frequently represented as a partially observed system. Consequently, the learning system must approximate the real state using filters. Experience with an actual physical system is time-consuming, costly, and difficult to duplicate. Because each trial run is expensive, such applications drive us to concentrate on issues that do not surface as frequently in traditional RL benchmark instances. Appropriate approximations of state, policy, value function, and/or system dynamics must be introduced in order to learn within a tolerable time period. While real-world experience is costly, it can typically not be substituted solely by simulation learning. Even little modeling flaws in analytical or learned models of the system might result in significantly divergent behavior, at least for highly dynamic jobs. As a result, algorithms must be resistant to under-modeling and uncertainty.

Another issue that arises frequently in robotic RL is generating appropriate reward functions. To cope with the expense of real-world experience, rewards that steer the learning system fast to success are required. This problem is known as reward shaping, and it requires a significant amount of manual contribution<sup>[148]</sup>. In robotics, defining excellent reward functions necessitates a substantial degree of domain expertise and can be difficult in practice.

Not all RL methods are equally appropriate for robotics. Indeed, many of the methods used to solve complex issues thus far have been model-based, and robot learning systems frequently use policy search



**Figure 10.** Main DL architectures. (A) Schematic diagram of RBMs; (B) schematic diagram of AEs; (C) schematic diagram of DBNs; (D) conceptual structure of CNNs. RBMs: restricted Boltzmann machines; DBNs: deep belief networks; AE: autoencoder; CNNs: convolutional neural networks.

methods rather than value function-based approaches. Such design decisions are in stark contrast to maybe the majority of early ML research. The papers that follow will discuss several approaches to incorporating

RL into robotics and manipulation. Kober *et al.*<sup>[149]</sup> conducted a comprehensive review of RL in robotics in 2013. They provide a reasonably comprehensive overview of “Real” Robotic RL and mention the most innovative studies, which are organized by significant findings.

In the last 15 years or so, the use of RL in robots has continuously risen. An overview of the RL-based implementation in robots’ control is shown in [Table 5](#)<sup>[150-172]</sup>, where each of the undermentioned papers has been categorized based on the nature of their approach.

A stacked Q-learning technique for a robot interacting with its surroundings was introduced by Digney<sup>[150]</sup>. In an inverted pole-balancing issue, Schaal<sup>[151]</sup> employed RL for robot learning. For compliance tasks, Kuan and Young<sup>[152]</sup> developed an RL-based mechanism in conjunction with a robust sliding mode impedance controller, which they evaluated in simulation. To cope with the variation in the different compliance tasks, they apply an RL-based method in their research. Bucak and Zohdy<sup>[153,154]</sup> proposed an RL-based control strategy for one and two link robots in 1999 and 2001. Althoefer *et al.*<sup>[155]</sup> used RL to attain motion and avoid obstacles in a Fuzzy rule-based system for a robot manipulator. Q-learning for robot control was investigated by Gaskett<sup>[156]</sup>. For a mobile robot navigation challenge, Smart and Kaelbling also opted for an RL-based approach<sup>[157]</sup>. For optimal control of a musculoskeletal-type robot arm with two joints and six muscles, Izawa *et al.*<sup>[158]</sup> used an RL actor-critic framework. For an optimum reaching task, they employed the proposed technique. RL approaches in humanoid robots are characterized, by Peters *et al.*<sup>[159]</sup>, as greedy methods, “vanilla” policy gradient methods, and natural gradient methods. They highly encourage the adoption of a natural gradient approach to control humanoid robots, because natural-actor-critic (NAC) structures converge fast and are better suited to high-dimensional systems like humanoid robots. They have proposed a number of different ways to design RL-based control systems for humanoid robots. An expansion of this study was given in 2009 by Bhatnagar *et al.*<sup>[160]</sup>. Theodorou *et al.*<sup>[161]</sup> employed RL for optimal control of arm kinematics. NAC applications in robotics were presented by Peters and Schaal<sup>[162]</sup>. For the estimate, the NAC employs the natural gradient approach. Other works presented here<sup>[163-165]</sup> go into greater depth on actor-critic based RL in robots. Buchli *et al.*<sup>[166]</sup> propose RL for variable impedance management methods based on policy improvement using a route integral approach. Only simulations were used to illustrate the efficiency of the suggested method. Theodorou *et al.*<sup>[167]</sup> used a robot dog to evaluate RL based on policy improvement using path integral<sup>[168]</sup>. RL-based control for robot manipulators in uncertain circumstances was given by Shah and Gopal<sup>[169]</sup>. Kim *et al.*<sup>[170,171]</sup> applied an RL-based method to determine acceptable compliance for various scenarios by interaction with the environment. The usefulness of Kim *et al.*<sup>[170,171]</sup>’s RL-based impedance learning technique has been demonstrated in simulations.

For a robot goalkeeper and inverted pendulum examples, Adam *et al.*<sup>[172]</sup> proposed a very interesting article on the experimental implementation of experience replay Q-learning and experience replay SARSA approaches. In this form of RL scheme, the data obtained during the online learning process is saved and fed back to the RL system continuously<sup>[172]</sup>. The results are encouraging, albeit the implementation method may not be appropriate for all actual systems, as the exploration phase indicates very irregular, nearly unstable behavior, which might harm a more delicate plant.

It is worth noting that several of the RL systems outlined above are conceptually well-developed, with convergence proofs available. However, there is still much work to be done on RL, and real-time implementations of most of these systems are still a great difficulty. Furthermore, adequate benchmark challenges<sup>[173]</sup> are required to test newly created or improved RL algorithms.

**Table 5. RL-based control in robotic control - an overview**

Approach	Employed by...
Q-learning	Digney <sup>[150]</sup> (1996), Gaskett <sup>[156]</sup> (2002), Shah and Gopal <sup>[169]</sup> (2009) and Adam et al. <sup>[172]</sup> (2012)
Optimal control/bio-mimetic learning	Izawa et al. <sup>[158]</sup> (2002) and Theodorou et al. <sup>[161]</sup> (2007)
NAC	Atkeson and Schaal <sup>[163]</sup> (1997), Peters et al. <sup>[159]</sup> (2003), Peters and Schaal <sup>[162]</sup> (2008), Hoffmann et al. <sup>[164]</sup> (2008) and Peters and Schaal <sup>[165]</sup> (2008)
Inverted pole-balancing	Schaal <sup>[151]</sup> (1996) and Adam et al. <sup>[172]</sup> (2012)
Impedance control	Kuan and Young <sup>[152]</sup> (1998) and Buchli et al. <sup>[166]</sup> (2010)
Fuzzy rule-based system	Althoefer et al. <sup>[155]</sup> (2001)
Navigation challenge	Smart and Kaelbling <sup>[157]</sup> (2002)
Route integral control	Buchli et al. <sup>[166]</sup> (2010)
Path integral	Theodorou et al. <sup>[167]</sup> (2010)

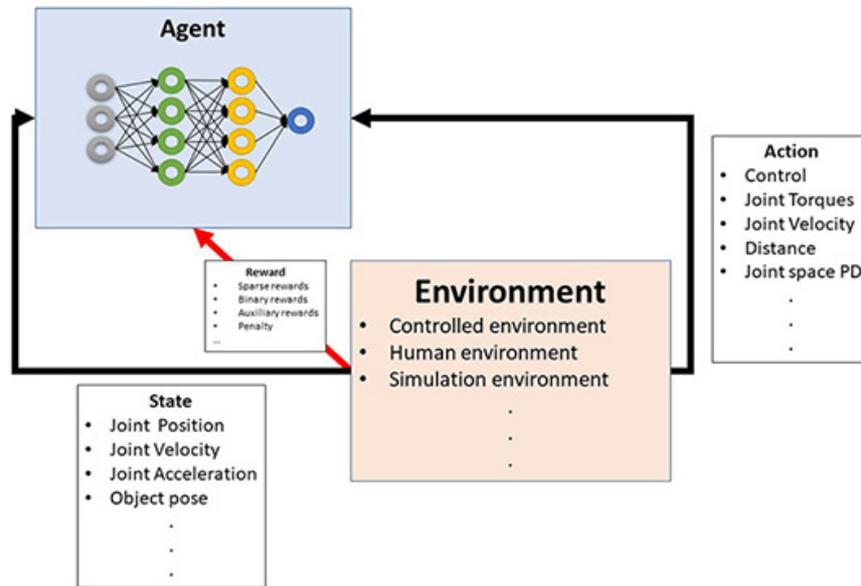
RL: Reinforcement learning; NAC: natural-actor-critic.

#### 4.2. Deep reinforcement learning for robotic manipulation control

In 2012, deep learning (DL) achieved its first major breakthrough with a CNN for classification<sup>[174]</sup>. It iteratively trains the parameters using loss computation and BP using hundreds of thousands of data-label pairs. Although this approach has developed steadily since its inception and is currently one of the most widely used DL structures, it is not ideal for robotic manipulation control because obtaining a large number of pictures of joint angles with labeled data to train the model is too time-consuming. CNN has been used in several studies to learn the motor torques required to drive a robot using raw RGB video pictures<sup>[175]</sup>. However, as we will see later, employing deep reinforcement learning (DRL) is a more promising and fascinating notion.

In the context of robotic manipulation control, the purpose of DRL is to train a deep policy NN, such as the one shown in [Figure 10](#), to discover the best command sequence for completing the job. The present state, as shown in [Figure 11](#), is the input, which can comprise the angles of the manipulator's joints, the location of the end effector, and their derivative information, such as velocity and acceleration. Furthermore, the current posture of target objects, as well as the status of relevant sensors if any are present in the surroundings, can be tallied in the current state. The policy network's output is an action that specifies which control instructions, such as torques or velocity commands, should be applied to each actuator. A positive reward will be produced when the robotic manipulator completes a job. The algorithm is supposed to discover the best successful control method for robotic manipulation using these delayed and weak data.

The study of sample efficiency for supervised deep learning determines the scale of the training set required in learning. Consequently, even though it is more challenging than supervised deep learning, the study of sample efficiency for DRL in robotic control provides how much data is needed to build an optimal policy. The first demonstration of using DRL on a robot was in 2015, when Levine et al.<sup>[176]</sup> applied trajectory optimization techniques and policy search methods with NNs to accomplish a practical sample efficient learning. They employ a recently developed policy search approach to learn a variety of dynamic manipulation behaviors with very broad policy representations, without requiring known models or example demonstrations in this study. This method uses repeatedly refitted time-varying linear models to train a collection of trajectories for the desired motion skill, and then unifies these trajectories into a single control policy that can generalize to new scenarios. Some modifications are needed in order to lower the sample count and automate parameter selection to enable this technique to run on a real robot. Finally, this approach has proven that the learning of robust controllers for complexity is possible, which did achieve various compound tasks such as stacking tight-fitting Lego blocks and putting together a toy airplane after



**Figure 11.** DRL scheme for Robotic manipulation control. DRL: Deep reinforcement learning.

minutes of interaction time.

The concept of imitation learning became very popular for robotic manipulation, since relying on learning from trial and error required a significant amount of system interaction time if based solely on DRL approaches<sup>[177]</sup>. In 2018, an interesting approach was proposed by Vecerik *et al.*<sup>[178]</sup> combining both imitation learning and task-reward-based learning, which improved the agent's abilities in simulation. The approach was based on an extension of Deep Deterministic Policy Gradient (DDPG) algorithm for tasks with sparse rewards. Unfortunately, in real robot experiments, the location of the object, as well as the explicit states of joints like position and velocity, must be specified, which limits the approach's applications to high-dimensional data<sup>[179]</sup>.

In 2017, Andrychowicz *et al.*<sup>[180]</sup> proposed Hindsight Experience Replay as a novel technique that provides for sample-efficient learning from sparse and binary rewards, avoiding the need for complex reward engineering. It may be used in conjunction with any off-policy RL algorithm to create an implicit curriculum.

In October 2021, AI researchers at Stanford University presented a new technique called deep evolutionary reinforcement learning, or DERL<sup>[181]</sup>. The new method employs a sophisticated virtual environment as well as RL to develop virtual agents that can change their physical form as well as their learning abilities. The discoveries might have far-reaching ramifications for AI research in general and robotics research in particular in the future. Each agent in the DERL architecture employs DRL to gain the abilities it needs to achieve its objectives throughout the course of its existence. MuJoCo, a virtual environment that enables very accurate rigid-body physics modeling, was employed by the researchers to create their framework. Universal Animal is their design space, and the objective is to construct morphologies that can master locomotion and item manipulation tasks in a range of terrains. The developed agents were put through their paces in eight various tasks, including patrolling, fleeing, manipulating items, and exploring. Their findings reveal that AI agents who have developed in different terrains learn and perform better than AI agents who

have only seen flat terrain.

An overview of the connection of the above-mentioned work is presented in [Table 6](#). Some basic problems are listed in the table, and each paper's approach is presented and categorized based on observation and action space, reward shaping and algorithm types.

Although DRL-based robotic manipulation control algorithms have proliferated in recent years, the issues of acquiring robust and diverse manipulation abilities for robots using DRL have yet to be properly overcome for real-world applications.

#### 4.3. Summary

Over the last several years, the robotics community has been progressively using RL and DRL-based algorithms to manage complicated robots or multi-robot systems, as well as to give end-to-end policies from perception to control. Since both algorithms base their knowledge acquisition on trial-and-error, they naturally require a large number of episodes, which limits the learning in terms of time and experience variability in real-world scenarios. In addition, the real-world experience must consider the potential dangers or unexpected behaviors of the considered robot, especially when it comes to safety-critical applications. Even though there are some successful real applications to DRL in robotics, especially with tasks involving object manipulations<sup>[182,183]</sup>, the success of its algorithms beyond the simulated worlds is fairly limited. Transferring DRL policies from simulation environments to reality, referred to as “sim-to-real”, is a necessary step toward more complex robotic systems that have DL-defined controllers. This has led to an increase in research in “sim-to-real” transfer, which resulted in many publications over the past few years.

Another angle that we see crucial for robotics applications is local vs. global learning. For instance, when humans learn a new task, like walking, they automatically build upon the previously learned skill in order to learn a new one, like running, which becomes significantly easier. It is essential to reuse other locally learned information from past data sets. When it comes to robot RL/DRL, the publicity of the making of such data sets with many skills should be available and accessible to everyone in robotic research, which would be considered a huge asset. When it comes to reward shaping, RL approaches have significantly benefited from it by using rewards that convey closeness and are not only based on binary success or failure. For robotics, it is challenging to shape such a reward design, hence, it would be optimal if the reward-shaping is physically motivated, like for instance, minimizing the torques while achieving a task.

## 5. CONCLUSION

In this review paper, we have surveyed the evolution of adaptive learning for nonlinear dynamic systems. In an initial step, after we introduced adaptive controllers and the modification techniques to overcome bounded disturbances, we have concluded that adaptive controllers have proven their effectiveness, especially in the processes that can be modeled linearly with slowly time-varying parameters relative to the system's dynamics. However, they do not provide stability for systems where parameter dynamics are at least the same magnitude as the system's dynamics.

In an evolutionary manner, AI-based techniques have emerged to improve the controller robustness. Newer methods, such as fuzzy logic and NNs were introduced. Essentially, these methods approximate a nonlinear function and provide a good representation of the nonlinear unknown plant, although it is typically used as a model-free controller. The plant is treated as a “black box”, with input and output data gathered and trained on. The AI framework addresses the plant's model after the training phase, and can handle the plant with practically no need for a mathematical model. It is feasible to build the complete algorithm using AI

**Table 6. DRL for robotic manipulation categorized by state and action space, algorithm and reward design**

State space	Action space	Algorithm type	Reward shaping
<b>Levine <i>et al.</i><sup>[176]</sup> (2015)</b>			
Joint angles and velocities	Joint torque	Trajectory optimization algorithm.	A penalty term is shaped as the sum of a quadratic term, and a Lorentzian $p$ -function. The first term encourages speed while the second term encourages precision. In addition, a quadratic penalty is applied to joint velocities and torques to smooth and control motions.
<b>Andrychowicz <i>et al.</i><sup>[180]</sup> (2017)</b>			
Joint angles & velocities + Objects' positions, rotations & velocities	4D action space. The first three are position related, the last one specifies the desired distance.	HER combined with any off-policy RL algorithm, like DDPG.	Binary and sparse rewards.
<b>Vecerik <i>et al.</i><sup>[178]</sup> (2018)</b>			
Joint position and velocity, joint torque, and global pose of the socket and plug.	Joint velocities.	An off-policy RL algorithm, called DDPGfD, is based on imitation learning.	First is a sparse reward function: +10 if the plug is within a small tolerance of the goal. The second reward is shaped by two terms: a reaching phase for alignment and an inserting phase to reach the goal.
<b>Gupta <i>et al.</i><sup>[181]</sup> (2021)</b>			
Depends on the agent morphology and include joint angles, angular velocities, readings of a velocimeter, accelerometer, and a gyroscope positioned at the head, and touch sensors attached to the limbs and head.	Chosen via a stochastic policy determined by the parameters of a deep NN that are learned via proximal policy optimization (PPO).	DERL, which is a simple computational framework operating by mimicking the intertwined processes of Darwinian evolution.	Two reward components. First relative to velocity and second relative to actuators' input.

DRL: Deep reinforcement learning; HER: Hindsight Experience Replay; DDPG: Deep Deterministic Policy Gradient.

techniques, or to merge the analytical and AI approaches such that some functions are done analytically and the remainder are performed using AI techniques.

We then briefly presented RL and DRL before we surveyed the previous work implementing both techniques in robot manipulation specifically. From this overview, it was clear that RL and DRL for robotics are not ready to offer a straightforward task yet. Although both techniques have evolved rapidly over the past few years with a wide range of applications, there is still a huge gap between theory and practice. The discrepancy between what we intend to solve and what we solve in practice, and accurately explaining the differences and how this affects our solution, we believe, is one of the core difficulties that plague the RL/DRL research community.

As RL/DRL researchers, we should take a step back and concentrate on the basics. By concentrating on the basics, we imply concentrating on simple, analyzable domains from which we may draw useful conclusions about the algorithms. Above all, areas in which we know what the best possible reward is. We hope that our survey helps the nonlinear dynamic control community in general, and the robotics community in particular, to quickly learn about this topic and become closely familiar with the current work being done and what work remains to be done. We also hope to assist researchers in deriving some conclusions from work carried out so far and provide them with new avenues for future research.

## DECLARATIONS

### Authors' contributions

Made substantial contributions to the conception and design of the article and interpreting the relevant literature: Harib M

Performed oversight and leadership responsibility for the activity planning and execution, as well as developed ideas and evolution of overarching aims: Chaoui H

Performed critical review, commentary and revision, as well as provided administrative, technical, and material support: Chaoui H, Miah S

### Availability of data and materials

Not applicable.

### Financial support and sponsorship

None.

### Conflicts of interest

All authors declared that there are no conflicts of interest.

### Ethical approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Copyright

© The Author(s) 2022.

## REFERENCES

1. Asetline J, Mancini A, Sarture C. A survey of adaptive control systems. *IRE Trans Automat Contr* 1958;6:102-8. [DOI](#)
2. Stromer PR. Adaptive or self-optimizing control systems - a bibliography. *IRE Trans Automat Contr* 1959;AC-4:65-8. [DOI](#)
3. Mishkin E, Ludwig BJ. Adaptive control systems. 1st ed. New York: McGraw-Hill; 1961.
4. Truxal JG. Adaptive control. *IFAC Proceedings Volumes* 1963;1:386-92. [DOI](#)
5. Eveleigh VW. Adaptive control and optimization technique. 1st ed. New York: McGraw-Hill; 1967.
6. Wittenmark B. Stochastic adaptive control methods: a survey. *Int J Control* 1975;21:705-30. [DOI](#)
7. Åström K, Borisson U, Ljung L, Wittenmark B. Theory and applications of self-tuning regulators. *Automatica* 1977;13:457-76. [DOI](#)
8. Åström K. Theory and applications of adaptive control - a survey. *Automatica* 1983;19:471-86. [DOI](#)
9. Jamali H. Adaptive control methods for mechanical manipulators: a comparative study. Monterey, CA: Naval Postgraduate School; 1989.
10. Mathelin MD, Lozano R. Robust adaptive identification of slowly time-varying parameters with bounded disturbances. *Automatica* 1999;35:1291-305. [DOI](#)
11. Deisenroth MP, Rasmussen CE. PILCO: a model-based and data-efficient approach to policy search. Proceedings of the 28th International Conference on International Conference on Machine Learning; 2011 Jun; Madison, WI, USA. 2011. p. 465-72.
12. Wang LY, Zhang JF. Fundamental limitations and differences of robust and adaptive control. Proceedings of the 2001 American

- Control Conference. (Cat. No.01CH37148); 2001 Jun 25-27; Arlington, VA, USA. IEEE; 2001. p. 4802-7. DOI
13. Ioannou PA, Sun J. Robust adaptive control. Mineola, NY: Courier Corporation; 2012.
  14. Lavretsky E. Adaptive output feedback design using asymptotic properties of LQG/LTR controllers. *IEEE Trans Automat Contr* 2012;57:1587-91. DOI
  15. Sastry S, Bodson M. adaptive control: stability, convergence and robustness. Mineola, NY: Dover Publications; 2011. DOI
  16. Larminat P. On overall stability of certain adaptive control systems. *IFAC Proceedings Volumes* 1979;12:1153-9. DOI
  17. Narendra K, Yuan-hao Lin. Stable discrete adaptive control. *IEEE Trans Automat Contr* 1980;25:456-61. DOI
  18. Peterson B, Narendra K. Bounded error adaptive control. *IEEE Trans Automat Contr* 1982;27:1161-8. DOI
  19. Fuchs J. Discrete adaptive control: a sufficient condition for stability and applications. *IEEE Trans Automat Contr* 1980;25:940-6. DOI
  20. Goodwin G, Ramadge P, Caines P. Discrete-time multivariable adaptive control. *IEEE Trans Automat Contr* 1980;25:449-56. DOI PubMed
  21. Egardt B. Global stability analysis of adaptive control systems with disturbances. Proceedings of the 1980 Joint Automatic Control Conference; 2021 Nov 1; San Fransisco, CA. 1980. DOI
  22. Rohrs CE, Valavani L, Athans M, Stein G. Robustness of adaptive control algorithms in the presence of unmodeled dynamics. 1982 21st IEEE Conference on Decision and Control; 1982 Dec 8-10; Orlando, FL, USA. IEEE; 1982. p. 3-11. DOI
  23. Aström KJ. Analysis of Rohrs counterexamples to adaptive control. The 22nd IEEE Conference on Decision and Control; 1983 Dec; San Antonio, TX, USA. 1983. p. 982-7. DOI
  24. Riedle B, Cyr B, Kokotovic P. Disturbance instabilities in an adaptive system. *IEEE Trans Automat Contr* 1984;29:822-4. DOI
  25. Ioannou P, Kokotovic P. Instability analysis and improvement of robustness of adaptive control. *Automatica* 1984;20:583-94. DOI
  26. Egardt B. Stability of adaptive controllers. Berlin Heidelberg: Springer; 1979. DOI
  27. Kreisselmeier G, Narendra K. Stable model reference adaptive control in the presence of bounded disturbances. *IEEE Trans Automat Contr* 1982;27:1169-75. DOI
  28. Samson C. Stability analysis of adaptively controlled systems subject to bounded disturbances. *Automatica* 1983;19:81-6. DOI
  29. Ioannou PA, Kokotovic PV. Adaptive systems with reduced models. New York, NY, USA: Springer-Verlag; 1983. DOI
  30. Peterson B, Narendra K. Bounded error adaptive control. *IEEE Trans Automat Contr* 1982;27:1161-8. DOI
  31. Narendra K, Annaswamy A. Robust adaptive control in the presence of bounded disturbances. *IEEE Trans Automat Contr* 1986;31:306-15. DOI
  32. Slotine JJE, Li W. Applied nonlinear control. Englewood Cliffs, NJ: Prentice Hall; 1991.
  33. Bunich AL. Rapidly converging algorithm for the identification of a linear system with limited noise. *Autom Remote Control* 1983;44:1049-54.
  34. Sastry SS. Model-reference adaptive control - stability, parameter convergence, and robustness. *IMA J Math Control Info* 1984;1:27-66. DOI
  35. Slotine JE, Coetsee JA. Adaptive sliding controller synthesis for non-linear systems. *International Journal of Control* 1986;43:1631-51. DOI
  36. . Adaptive control in the presence of disturbances. In: Ioannou PA, Kokotovic PV, editors. Adaptive systems with reduced models. Berlin/Heidelberg: Springer-Verlag; 1983. p. 81-90. DOI
  37. Ioannou P, Tsakalis K. A robust direct adaptive controller. *IEEE Trans Automat Contr* 1986;31:1033-43. DOI
  38. Ioannou P. Robust adaptive controller with zero residual tracking errors. *IEEE Trans Automat Contr* 1986;31:773-6. DOI
  39. Ioannou P. Robust direct adaptive control. The 23rd IEEE Conference on Decision and Control; 1984 Dec 12-14; Las Vegas, NV, USA. IEEE; 1984. p. 1015-20. DOI
  40. Tsakalis KS. The  $\sigma$ -modification in the adaptive control of linear time-varying plants. [1992] Proceedings of the 31st IEEE Conference on Decision and Control; 1992 Dec 16-18; Tucson, AZ, USA. IEEE; 1992. p. 694-8. DOI
  41. He Z, Huang D, Xu J. On the asymptotic property analysis for a class of adaptive control systems with  $\sigma$ -modification: adaptive control systems with  $\sigma$ -modification. *Int J Adapt Control Signal Process* 2013;27:620-34. DOI
  42. Li MY, Muldowney JS. A geometric approach to global-stability problems. *SIAM Journal on Mathematical Analysis* 1996;27:14. DOI
  43. Narendra K, Annaswamy A. A new adaptive law for robust adaptation without persistent excitation. *IEEE Trans Automat Contr* 1987;32:134-45. DOI
  44. Lasalle J. Some extensions of Liapunov's second method. *IRE Trans Circuit Theory* 1960;7:520-7. DOI
  45. Mattern DL. Practical applications and limitations of adaptive control. Available from: <http://www.proquest.com/docview/303617884/abstract/FC4A275C8474474PQ/1> [Last accessed on 8 Mar 2022].
  46. Kreisselmeier G, Anderson B. Robust model reference adaptive control. *IEEE Trans Automat Contr* 1986;31:127-33. DOI PubMed PMC
  47. Davidson JM. Model reference adaptive control specification for a steam heated finned tube heat exchanger. Available from: <https://www.proquest.com/docview/302770965/citation/9192D8E407D24AFBPQ/1> [Last accessed on 8 Mar 2022].
  48. Davison E, Taylor P, Wright J. On the application of tuning regulators to control a commercial heat exchanger. *IEEE Trans Automat Contr* 1980;25:361-75. DOI
  49. Harrell RC, Kranzler GA, Hsu CS. Adaptive control of the fluid heat exchange process. *J Dyn Syst Meas Control* 1987;109:49-52.

## DOI

50. Zhang Q, Tomizuka M. Multivariable direct adaptive control of thermal mixing processes. *J Dyn Syst Meas Control* 1985;107:278-83. DOI
51. Lukas MP, Kaya A. Adaptive control of a heat exchanger using function blocks. *Chemical Engineering Communications* 2007;24:259-73. DOI
52. Harris CJ, Billings SA. Self-tuning and adaptive control - theory and applications. 1st ed. London: Peter Peregrinus, Ltd; 1981.
53. Dubowsky S, Desforjes DT. The application of model-referenced adaptive control to robotic manipulators. *J Dyn Syst Meas Control* 1979;101:193-200. DOI
54. Dubowsky S. On the adaptive control of robotic manipulators: the discrete-time case. *IEEE Trans Automat Contr* 1981. DOI
55. Nicosia S, Tomei P. Model reference adaptive control algorithms for industrial robots. *Automatica* 1984;20:635-44. DOI
56. Koivo A, Guo TH. Adaptive linear controller for robotic manipulators. *IEEE Trans Automat Contr* 1983;28:162-71. DOI
57. Horowitz R, Tomizuka M. An adaptive control scheme for mechanical manipulators - compensation of nonlinearity and decoupling control. *J Dyn Syst Meas Control* 1986;108:127-35. DOI
58. Narendra KS, Parthasarathy K. Adaptive identification and control of dynamical systems using neural networks. Proceedings of the 28th IEEE Conference on Decision and Control; 1989 Dec 13-15; Tampa, FL, USA. 1989. p. 1737-8. DOI
59. Lee C. Fuzzy logic in control systems: fuzzy logic controller. II. *IEEE Trans Syst, Man, Cybern* 1990;20:419-35. DOI
60. Sutton RS, Barto AG, Williams RJ. Reinforcement learning is direct adaptive optimal control. *IEEE Control Syst* 1992;12:19-22. DOI
61. Yechiel O. A survey of adaptive control. *IRATJ* 2017;3:0053. DOI PubMed
62. Malik O. Amalgamation of adaptive control and AI techniques: applications to generator excitation control. *Annu Rev Control* 2004;28:97-106. DOI
63. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci U S A* 1982;79:2554-8. DOI PubMed PMC
64. Hopfield JJ, Tank DW. "Neural" computation of decisions in optimization problems. *Biol Cybern* 1985;52:141-52. DOI PubMed
65. Burr D. Experiments on neural net recognition of spoken and written text. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1162-8. DOI
66. Gorman R, Sejnowski T. Learned classification of sonar targets using a massively parallel network. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1135-40. DOI
67. Sejnowski T, Rosenberg CR. Parallel networks that learn to pronounce English text. *Complex Syst* 1987;1:145-68.
68. Widrow B, Winter R, Baxter R. Layered neural nets for pattern recognition. *IEEE Trans Acoust, Speech, Signal Processing* 1988;36:1109-18. DOI
69. Levin AU, Narendra KS. Control of nonlinear dynamical systems using neural networks: controllability and stabilization. *IEEE Trans Neural Netw* 1993;4:192-206. DOI PubMed
70. Narendra KS, Parthasarathy K. Identification and control of dynamical systems using neural networks. *IEEE Trans Neural Netw* 1990;1:4-27. DOI PubMed
71. Sontag ED. Feedback stabilization using two-hidden-layer nets. *IEEE Trans Neural Netw* 1992;3:981-90. DOI PubMed
72. Barto AG. Connectionist learning for control: an overview. In: Miller WT, Sutton RS, Werbos PJ. *Neural networks for control*. Cambridge, MA, USA: MIT Press; 1990. p. 5-58.
73. Dai SL, Wang C, Wang M. Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 2014;25:111-23. DOI PubMed
74. Chen CL, Liu YJ, Wen GX. Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems. *IEEE Trans Cybern* 2014;44:583-93. DOI PubMed
75. Dai S, Wang M, Wang C. Neural learning control of marine surface vessels with guaranteed transient tracking performance. *IEEE Trans Ind Electron* 2016;63:1717-27. DOI
76. Li H, Bai L, Wang L, Zhou Q, Wang H. Adaptive neural control of uncertain nonstrict-feedback stochastic nonlinear systems with output constraint and unknown dead zone. *IEEE Trans Syst Man Cybern, Syst* 2017;47:2048-59. DOI
77. Cheng L, Liu W, Hou Z, Yu J, Tan M. Neural-network-based nonlinear model predictive control for piezoelectric actuators. *IEEE Trans Ind Electron* 2015;62:7717-27. DOI
78. Ren B, Ge SS, Su CY, Lee TH. Adaptive neural control for a class of uncertain nonlinear systems in pure-feedback form with hysteresis input. *IEEE Trans Syst Man Cybern B Cybern* 2009;39:431-43. DOI PubMed
79. Luo B, Huang T, Wu HN, Yang X. Data-driven  $H_\infty$  control for nonlinear distributed parameter systems. *IEEE Trans Neural Netw Learn Syst* 2015;26:2949-61. DOI PubMed
80. Liu Y, Tong S. Barrier Lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems. *Automatica* 2017;76:143-52. DOI
81. Li Y, Qiang S, Zhuang X, Kaynak O. Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. *IEEE Trans Neural Netw* 2004;15:693-701. DOI PubMed
82. Zhang H, Luo Y, Liu D. Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Trans Neural Netw* 2009;20:1490-503. DOI PubMed
83. Barto AG, Sutton RS, Anderson CW. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans*

- Syst, Man, Cybern* 1983;SMC-13:834-46. DOI
84. Michie D, Chambers RA. Boxes: an experiment in adaptive control. Edinburgh, UK: Oliver and Boyd; 1968. p. 137-52.
  85. Michie D, Chambers RA. Boxes' as a model of pattern-formation. 1st ed. Edinburgh: Edinburgh univ. press; 1968. p. 206-15. DOI
  86. Anderson CW. Strategy Learning with multilayer connectionist representations. proceedings of the fourth international workshop on machine learning. Elsevier; 1987. p. 103-14. DOI
  87. Anderson C. Learning to control an inverted pendulum using neural networks. *IEEE Control Syst Mag* 1989;9:31-7. DOI
  88. Lin CS, Kim H. CMAC-based adaptive critic self-learning control. *IEEE Trans Neural Netw* 1991;2:530-3. DOI PubMed
  89. Albus JS. Theoretical and experimental aspects of a Cerebellar Model. Available from: [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=820153](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=820153) [Last accessed on 8 Mar 2022].
  90. Albus JS. A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *J Dyn Syst Meas Control* 1975;97:220-7. DOI
  91. Albus JS. Mechanisms of planning and problem solving in the brain. *Math Biosci* 1979;45:247-93. DOI
  92. Albus JS. Brains, behavior, and robotics. 1st ed. Peterborough: BYTE Books; 1981.
  93. Huang, Chien-lo Huang. Control of an inverted pendulum using grey prediction model. *IEEE Trans on Ind Applicat* 2000;36:452-8. DOI
  94. Pathak K, Franch J, Agrawal S. Velocity and position control of a wheeled inverted pendulum by partial feedback linearization. *IEEE Trans Robot* 2005;21:505-13. DOI
  95. Li, Jun Luo. Adaptive Robust dynamic balance and motion controls of mobile wheeled inverted pendulums. *IEEE Trans Contr Syst Technol* 2009;17:233-41. DOI
  96. Chaoui H, Gueaieb W, Yagoub MCE. ANN-based adaptive motion and posture control of an inverted pendulum with unknown dynamics. 2009 3rd International Conference on Signals, Circuits and Systems (SCS); 2009 Nov 6-8; Medenine, Tunisia. IEEE; 2009. p. 1-6. DOI
  97. Guez A, Ahmad Z. Solution to the inverse kinematics problem in robotics by neural networks. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 617-24. DOI
  98. . Elsley. A learning architecture for control based on back-propagation neural networks. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 587-94. DOI
  99. Jamshidi M, Horne B, Vadiiee N. A neural network-based controller for a two-link robot. 29th IEEE Conference on Decision and Control; 1990 Dec 5-7; Honolulu, HI, USA. IEEE; 1990. p. 3256-7. DOI
  100. Karakasoglu A, Sundareshan MK. Decentralized variable structure control of robotic manipulators: neural computational algorithms. 29th IEEE Conference on Decision and Control; 1990 Dec 5-7; Honolulu, HI, USA. IEEE; 1990. p. 3258-9. DOI
  101. Xu G, Scherrer H, Schweitzer G. Application of neural networks on robot grippers. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 337-42. DOI
  102. Wilhelmsen K, Cotter N. Neural network based controllers for a single-degree-of-freedom robotic arm. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 407-13. DOI
  103. Miller WT, Glanz FH, Kraft LG. Application of a general learning algorithm to the control of robotic manipulators. *Int J Rob Res* 1987;6:84-98. DOI
  104. Miller W. Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE J Robot Automat* 1987;3:157-65. DOI
  105. Miller WT. Real time learned sensor processing and motor control for a robot with vision. *Neural Networks* 1988;1:347. DOI
  106. Miller WT, Hewes RP. Real time experiments in neural network based learning control during high speed nonrepetitive robotic operations. Proceedings IEEE International Symposium on Intelligent Control 1988; 1988 Aug 24-26; Arlington, VA, USA. IEEE; 1988. p. 513-8. DOI
  107. Miller W. Real-time application of neural networks for sensor-based control of robots with vision. *IEEE Trans Syst, Man, Cybern* 1989;19:825-31. DOI
  108. Miller W, Glanz F, Kraft L. CMAC: an associative neural network alternative to backpropagation. *Proc IEEE* 1990;78:1561-7. DOI
  109. Huan L, Iberall, Bekey. Building a generic architecture for robot hand control. IEEE 1988 International Conference on Neural Networks; 1988 Jul 24-27; San Diego, CA, USA. IEEE; 1988. p. 567-74. DOI
  110. Wang SD, Yeh HMS. Self-adaptive neural architectures for control applications. 1990 IJCNN International Joint Conference on Neural Networks; 1990 Jun 17-21; San Diego, CA, USA. IEEE; 1990. p. 309-14. DOI
  111. Seidl D, Lam SL, Putman J, Lorenz R. Neural network compensation of gear backlash hysteresis in position-controlled mechanisms. *IEEE Trans on Ind Applicat* 1995;31:1475-83. DOI
  112. Olsson H, Åström K, Canudas de Wit C, Gäfvert M, Lischinsky P. Friction models and friction compensation. *European Journal of Control* 1998;4:176-95. DOI
  113. Katsura S, Suzuki J, Ohnishi K. Pushing operation by flexible manipulator taking environmental information into account. *IEEE Trans Ind Electron* 2006;53:1688-97. DOI
  114. Katsura S, Ohnishi K. Force servoing by flexible manipulator based on resonance ratio control. *IEEE Trans Ind Electron* 2007;54:539-47. DOI
  115. Ghorbel F, Hung J, Spong M. Adaptive control of flexible-joint manipulators. *IEEE Control Syst Mag* 1989;9:9-13. DOI
  116. Chien M, Huang A. Adaptive control for flexible-Joint electrically driven robot with time-varying uncertainties. *IEEE Trans Ind*

- Electron* 2007;54:1032-8. DOI
117. Hauschild JP, Heppner GR. Control of harmonic drive motor actuated flexible linkages. Proceedings 2007 IEEE International Conference on Robotics and Automation; 2007 Apr 10-14; Rome, Italy. IEEE; 2007. p. 3451-6. DOI
  118. Kong K, Tomizuka M, Moon H, Hwang B, Jeon D. Mechanical design and impedance compensation of SUBAR (Sogang University's Biomedical Assist Robot). 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics; 2008 Jul 2-5; Xi'an, China. IEEE; 2008. p. 377-82. DOI
  119. Ghorbel F, Spong MW. Adaptive integral manifold control of flexible joint robot manipulators. Proceedings 1992 IEEE International Conference on Robotics and Automation; 1992 May 12-14; Nice, France. IEEE; 1992. p. 707-14. DOI
  120. Al-ashoor R, Patel R, Khorasani K. Robust adaptive controller design and stability analysis for flexible-joint manipulators. *IEEE Trans Syst, Man, Cybern* 1993;23:589-602. DOI
  121. Ott C, Albu-Schaffer A, Hirzinger G. Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2002 Sep 30-Oct 4; Lausanne, Switzerland. IEEE; 2002. p. 2018-24. DOI
  122. Spong MW. Modeling and control of elastic joint robots. *J Dyn Syst Meas Control* 1987;109:310-8. DOI
  123. Ge SS, Postlethwaite I. Adaptive neural network controller design for flexible joint robots using singular perturbation technique. *Transactions of the Institute of Measurement and Control* 1995;17:120-31. DOI
  124. Taghirad HD, Khosravi MA. Design and simulation of robust composite controllers for flexible joint robots. 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422); 2003 Sep 14-19; Taipei, Taiwan. IEEE; 2003. p. 3108-13. DOI
  125. Huang L, Ge SS, Lee TH. Adaptive position/force control of an uncertain constrained flexible joint robots - singular perturbation approach. SICE 2004 Annual Conference; 2004 Aug 4-6; Sapporo, Japan; 2004. p. 220-5.
  126. Chaoui H, Gueaieb W. Type-2 fuzzy logic control of a flexible-joint manipulator. *J Intell Robot Syst* 2008;51:159-86. DOI PubMed
  127. Karray F, Gueaieb W, Al-Sharhan S. The hierarchical expert tuning of PID controllers using tools of soft computing. *IEEE Trans Syst Man Cybern B Cybern* 2002;32:77-90. DOI PubMed
  128. Gueaieb W, Karray F, Al-sharhan S. A robust adaptive fuzzy position/force control scheme for cooperative manipulators. *IEEE Trans Contr Syst Technol* 2003;11:516-28. DOI
  129. Kim E. Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic. *IEEE Trans Fuzzy Syst* 2004;12:368-78. DOI
  130. Chaoui H, Gueaieb W, Yagoub MCE, Sicard P. Hybrid neural fuzzy sliding mode control of flexible-joint manipulators with unknown dynamics. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics; 2006 Nov 6-10; Paris, France. IEEE; 2006. p. 4082-7. DOI
  131. Chaoui H, Sicard P, Lakhsasi A. Reference model supervisory loop for neural network based adaptive control of a flexible joint with hard nonlinearities. Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513); 2004 May 2-5; Niagara Falls, ON, Canada. IEEE; 2004. p. 2029-34. DOI
  132. Chaoui H, Sicard P, Lakhsasi A, Schwartz H. Neural network based model reference adaptive control structure for a flexible joint with hard nonlinearities. 2004 IEEE International Symposium on Industrial Electronics; 2004 May 4-7; Ajaccio, France. IEEE; 2004. p. 271-6. DOI
  133. Hui, Fuchun S, Zengqi S. Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks. *J Intell Robot Syst* 2002;34:453-66. DOI
  134. Subudhi B, Morris AS. Singular perturbation based neuro-H<sub>∞</sub> control scheme for a manipulator with flexible links and joints. *Robotica* 2006;24:151-61. DOI
  135. Chaoui H, Sicard P, Gueaieb W. ANN-based adaptive control of robotic manipulators with friction and joint elasticity. *IEEE Trans Ind Electron* 2009;56:3174-87. DOI
  136. Hou ZG, Cheng L, Tan M. Multicriteria optimization for coordination of redundant robots using a dual neural network. *IEEE Trans Syst Man Cybern B Cybern* 2010;40:1075-87. DOI PubMed
  137. Li Z, Su CY. Neural-adaptive control of single-master-multiple-slaves teleoperation for coordinated multiple mobile manipulators with time-varying communication delays and input uncertainties. *IEEE Trans Neural Netw Learn Syst* 2013;24:1400-13. DOI PubMed
  138. Li Z, Xia Y, Sun F. Adaptive fuzzy control for multilateral cooperative teleoperation of multiple robotic manipulators under random network-induced delays. *IEEE Trans Fuzzy Syst* 2014;22:437-50. DOI
  139. He W, Huang H, Ge SS. Adaptive neural network control of a robotic manipulator with time-varying output constraints. *IEEE Trans Cybern* 2017;47:3136-47. DOI PubMed
  140. He W, Ouyang Y, Hong J. Vibration control of a flexible robotic manipulator in the presence of input deadzone. *IEEE Trans Ind Inf* 2017;13:48-59. DOI
  141. Zhu G, Ge S, Lee T. Simulation studies of tip tracking control of a single-link flexible robot based on a lumped model. *Robotica* 1999;17:71-8. DOI
  142. Sun C, He W, Hong J. Neural network control of a flexible robotic manipulator using the lumped spring-mass model. *IEEE Trans Syst Man Cybern, Syst* 2017;47:1863-74. DOI
  143. Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Belmont, MA: Athena Scientific; 1996.
  144. Pitts W, McCulloch WS. How we know universals; the perception of auditory and visual forms. *Bull Math Biophys* 1947;9:127-47.

[DOI PubMed](#)

145. Liu R. Multispectral images-based background subtraction using Codebook and deep learning approaches. Available from: <https://www.theses.fr/2020UBFCA013.pdf> [Last accessed on 8 Mar 2022].
146. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE. A survey of deep neural network architectures and their applications. *Neurocomputing* 2017;234:11-26. [DOI](#)
147. Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge. *Nature* 2017;550:354-9. [DOI PubMed](#)
148. Laud AD. Theory and application of reward shaping in reinforcement learning. Available from: <https://www.proquest.com/openview/bb29dc3d66ecbe7ab65560dd2c4147f/1?pq-origsite=gscholar&cbl=18750&diss=y> [Last accessed on 8 Mar 2022].
149. Kober J, Bagnell JA, Peters J. Reinforcement learning in robotics: a survey. *Int J Rob Res* 2013;32:1238-74. [DOI PubMed](#)
150. Digney BL. Nested Q-learning of hierarchical control structures. Proceedings of International Conference on Neural Networks (ICNN'96); 1996 Jun 3-6; Washington, DC, USA. IEEE; 1996. p. 161-6. [DOI](#)
151. Schaal S. Learning from demonstration. Proceedings of the 9th International Conference on Neural Information Processing Systems; 1996 Dec; Cambridge, MA, USA. IEEE; 1996. p. 1040-6. [DOI](#)
152. Kuan C, Young K. Reinforcement learning and robust control for robot compliance tasks. *J Intell Robot Syst* 1998;23:165-82. [DOI](#)
153. Bucak IO, Zohdy MA. Application of reinforcement learning control to a nonlinear dexterous robot. Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304); 1999 Dec 7-10; Phoenix, AZ, USA. IEEE; 1999. p. 5108-13. [DOI](#)
154. Bucak IO, Zohdy MA. Reinforcement learning control of nonlinear multi-link system. *Eng Appl Artif Intell* 2001;14:563-75. [DOI](#)
155. Althoefer K, Krekelberg B, Husmeier D, Seneviratne L. Reinforcement learning in a rule-based navigator for robotic manipulators. *Neurocomputing* 2001;37:51-70. [DOI](#)
156. Gaskett C. Q-learning for robot control. Available from: <https://digitalcollections.anu.edu.au/bitstream/1885/47080/5/01front.pdf> [Last accessed on 8 Mar 2022].
157. Smart WD, Kaelbling LP. Reinforcement learning for robot control. *Proc SPIE* 2002. [DOI](#)
158. Izawa J, Kondo T, Ito K. Biological robot arm motion through reinforcement learning. Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292); 2002 May 11-15; Washington, DC, USA. IEEE; 2002. p. 3398-403. [DOI](#)
159. Peters J, Vijayakumar S, Schaal S. Reinforcement learning for humanoid robotics. 3rd IEEE-RAS International Conference on Humanoid Robots; 2003 Sep 29-30; Karlsruhe, Germany. 2003.
160. Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M. Natural actor-critic algorithms. *Automatica* 2009;45:2471-82. [DOI](#)
161. Theodorou E, Peters J, Schaal S. Reinforcement learning for optimal control of arm movements. Poster presented at 37th Annual Meeting of the Society for Neuroscience (Neuroscience 2007); San Diego, CA, USA. 2007. [DOI](#)
162. Peters J, Schaal S. Natural actor-critic. *Neurocomputing* 2008;71:1180-90. [DOI](#)
163. Atkeson CG, Schaal S. Learning tasks from a single demonstration. Proceedings of International Conference on Robotics and Automation; 1997 Apr 25-25; Albuquerque, NM, USA. IEEE; 1997. p. 1706-12. [DOI](#)
164. Hoffmann H, Theodorou E, Schaal S. Behavioral experiments on reinforcement learning in human motor control. Available from: <https://www.researchgate.net/publication/325463394> [Last accessed on 8 Mar 2022].
165. Peters J, Schaal S. Learning to control in operational space. *Int J Rob Res* 2008;27:197-212. [DOI](#)
166. Buchli J, Theodorou E, Stulp F, Schaal S. Variable impedance control - a reinforcement learning approach. In: Matsuoka Y, Durrant-Whyte H, Neira J, editors. Robotics: Science and Systems VI. Cambridge: MIT Press; 2011. [DOI](#)
167. Theodorou E, Buchli J, Schaal S. Reinforcement learning of motor skills in high dimensions: a path integral approach. 2010 IEEE International Conference on Robotics and Automation; 2010 May 3-7; Anchorage, AK, USA. IEEE; 2010. p. 2397-403. [DOI](#)
168. Kappen HJ. Path integrals and symmetry breaking for optimal control theory. *J Stat Mech* 2005;2005:P11011. [DOI](#)
169. Shah H, Gopal M. Reinforcement learning control of robot manipulators in uncertain environments. 2009 IEEE International Conference on Industrial Technology; 2009 Feb 10-13; Churchill, VIC, Australia. IEEE; 2009. p. 1-6. [DOI](#)
170. Kim B, Kang B, Park S, Kang S. Learning robot stiffness for contact tasks using the natural actor-critic. 2008 IEEE International Conference on Robotics and Automation; 2008 May 19-23; Pasadena, CA, USA. IEEE; 2008. p. 3832-7. [DOI](#)
171. Kim B, Park J, Park S, Kang S. Impedance learning for robotic contact tasks using natural actor-critic algorithm. *IEEE Trans Syst Man Cybern B Cybern* 2010;40:433-43. [DOI PubMed](#)
172. Adam S, Busoni L, Babuska R. Experience replay for real-time reinforcement learning control. *IEEE Trans Syst , Man, Cybern C* 2012;42:201-12. [DOI](#)
173. Hafner R, Riedmiller M. Reinforcement learning in feedback control: Challenges and benchmarks from technical process control. *Mach Learn* 2011;84:137-69. [DOI](#)
174. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM* 2017;60:84-90. [DOI](#)
175. Levine S, Finn C, Darrell T, Abbeel P. End-to-end training of deep visuomotor policies. Available from: <http://arxiv.org/abs/1504.00702> [Last accessed on 8 Mar 2022].
176. Levine S, Wagener N, Abbeel P. Learning contact-rich manipulation skills with guided policy search. Available from: <http://arxiv.org/abs/1501.05611> [Last accessed on 8 Mar 2022].

177. Tai L, Zhang J, Liu M, Boedecker J, Burgard W. A survey of deep network solutions for learning control in robotics: from reinforcement to imitation. Available from: <http://arxiv.org/abs/1612.07139> [Last accessed on 8 Mar 2022].
178. Vecerik M, Hester T, Scholz J, et al. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. Available from: <http://arxiv.org/abs/1707.08817> [Last accessed on 8 Mar 2022].
179. Liu R, Nageotte F, Zanne P, de Mathelin M, Dresch-langley B. Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics* 2021;10:22. DOI
180. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. Available from: <https://arxiv.org/abs/1707.01495v3> [Last accessed on 8 Mar 2022].
181. Gupta A, Savarese S, Ganguli S, Fei-Fei L. Embodied intelligence via learning and evolution. *Nat Commun* 2021;12:5721. DOI PubMed PMC
182. Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. Available from: <http://arxiv.org/abs/1709.10087> [Last accessed on 8 Mar 2022].
183. Matas J, James S, Davison AJ. Sim-to-real reinforcement learning for deformable object manipulation. Available from: <http://arxiv.org/abs/1806.07851> [Last accessed on 8 Mar 2022].