A Review on the Feasibility of Artificial Intelligence in Mechatronics

Amin Hashemi and Mohammad Bagher Dowlatshahi

1 Introduction

Intelligent control strategies are very flexible for describing from an automation point of view. While the system performs critical functions, this concept is dynamically implemented in real-time. So it is different from traditional feedback. Thus, if the law of control is continuously updated, we can assume the classical adaptive control to be intelligent. This kind of system can be considered borderline according to the classification perspective [1]. If we trace intelligent control approaches from the source for mechatronics analysis, we will face analyzing and processing big data, the evolution of mathematic-based methods, and programming. The exponential growth process of this research area reached the early 2010s and did not stop declining [2].

Various artificial intelligence methods and areas are utilized in mechatronics and robotics, including artificial neural networks (ANNs), machine learning, evolutionary computing algorithms, and fuzzy logic. Machine learning consists of deep learning, reinforcement learning, classical learning (unsupervised and supervised), neural networks (NN), and ensemble methods. Intelligent control algorithms analyze large data sets and exploit beneficial patterns from actions taken by utilizing a variety of probabilistic, statistical, and optimization methods [2]. In the automatic control field, reinforcement learning is practical. Ensemble strategies and classical learning are also used for classifying and processing data sets against neural networks [3–5]. Conversely, neural networks are practical in dealing with unlabeled features and complex data.

A. Hashemi · M. B. Dowlatshahi (✉)
Department of Computer Engineering, Faculty of Engineering, Lorestan University, Khorramabad, Iran
e-mail: dowlatshahi.mb@lu.ac.ir

A. Hashemi
e-mail: hashemi.am@fe.lu.ac.ir

E. Momeni et al. (eds.), *Artificial Intelligence in Mechatronics and Civil Engineering*, Emerging Trends in Mechatronics, https://doi.org/10.1007/978-981-19-8790-8_3

In autonomous systems that interact with the real world, a critical challenge is the safety and reliability of utilizing intelligent control approaches. This problem is the subject of a review article [6] in which an asymptotic analysis of intelligent control approaches convergence is conducted.

This chapter discusses modern intelligent control approaches in mechatronics to recognize open issues and trends in intelligent control.

The chapter is constructed as follows: Sect. 2 recalls multiple intelligent control approaches. The applications of intelligent approaches in engineering control problems are presented in Sect. 3. The chapter is concluded in Sect. 4.

2 Smart Control Methods

Intelligent control is an apart discipline, but the application of new concepts, such as neural networks to control loops, utilizing different scientific approaches constructed based on automatic control theory, is required. Therefore, intelligent control can improve its performance every time by learning from previous experiences as a combined approach.

In the framework of intellectual approaches, assume the most general modern control theory methods. This chapter pays the most attention to machine learning since some of them are well-known, and there is no further explanation.

2.1 Adaptive Control Methods

Like optimal control, adaptive control is constructed based on a well-developed mathematical and theoretical justification [2]. This method became the initial step for intelligent control, as an integration of adaptive controllers within the framework of the classical automatic control theory provides a quality of operation of the system given the conditions of the parameters of the object and the specification of the external environment are unknown or change indefinitely. The adaptation principle can be considered the heart of intelligent control processes, which evolves from self-optimizing controllers to adaptive learning systems [7].

Adaptive control algorithms for time-discrete systems were applied to artificial intelligence by Yakubovich, who received several algorithms for training linear classification models [2, 8–10]. The Stripe Algorithm (SA) proposed by Yakubovich in a recently published article [11] has shown acceptable performance in machine learning. SA achieves higher performance than traditional linear learning methods by numerical analysis in online machine learning, making it suitable for this field. Lipkovich [12] provides various strategies for the reduction of loss optimization problems in dealing with inequalities systems, considering both regression and classification problems. In reference [11], a comparison analysis is conducted between

SA and modern linear analogs, including logistic and linear regression. Complex non-linear models have the potential to outperform SA. However, the last one includes the common points of interest of linear models, like explainability, development, and implementation [12]. It can be noted that the presence of a hypothesis does not ensure practical application, especially in experimental conditions of control systems [12].

2.2 Optimization Techniques

Optimization techniques emerged before machine learning historically and were utilized to discover the extrema of a function [13]. Most machine learning problems are based on the theory of optimality. This theory can be generally formulated as the minimization of multiple features J regarding multiple parameters $\theta = J(\theta) \to \min\limits_{\theta \epsilon X}$. The form of the minimized value is determined by the machine learning approach. As an example, the prediction error on the existing sample is minimized in a regression or classification problem, while the greatest advantage from the activities of the plant is discovered in reinforcement learning. This can be accomplished by any search algorithm. As you can see, there are many types, methods, and uses of mathematical optimization.

2.3 ANNs

ANNs inspire biological networks as powerful artificial intelligence tools. ANN is an object that imitates the neural network constituting the human brain so that the computer can learn and make decisions like a human. An input layer, a hidden layer or more, and nodes or neurons as numerous simple computational components as an output layer construct an ANN structure. A simple ANN with two inputs and two hidden layers is presented in Fig. 1. This additionally includes relationships between neurons in consecutive layers through the weights. These weights can change the signal sent from one node to another and increase or decrease the impact of a particular relationship. A weighted input plus one bias from each neuron in the previous layer is received by each hidden layer neuron. The output of neurons is determined by their activation function. An ANN structure is shown below

$$Y = f\left(\sum_{i=1}^{n} w_i v_i - b\right) \tag{1}$$

In Eq. 1, f refers to the activation function, v_i and w_i are shown the input values and the weights of neurons, respectively. Also, y refers to the network's output, b is the bias, and n indicates the neuron's number in the hidden layers. The performance

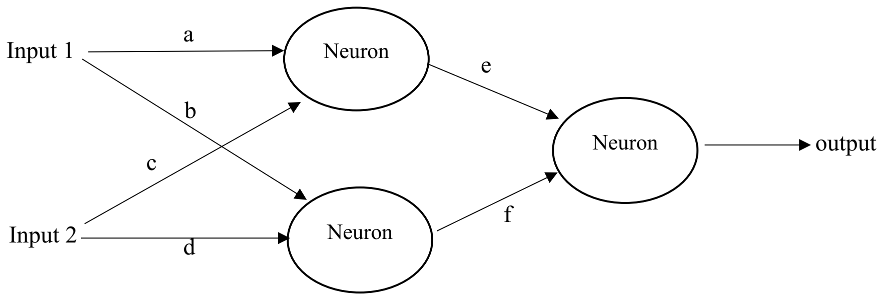

Fig. 1 A simple ANN structure

of the model can be enhanced by updating the network weights during the training phase. ANN's neuron weights determine how the input data affects output data. The primary weights are selected randomly [14].

The network's internal weights are tuned using a learning algorithm. Backpropagation (BP) algorithms are today's most common form of training in ANNs. Also, optimization methods such as genetic algorithm and particle swarm optimization are practical in optimizing the ANN [14].

Using NNs is effective for noisy and non-linear system controls, and adaptability is provided for the system. The NN can work in real time after training. The constant NS advancement in properties and structure aims to overcome the existing shortcomings. The heuristic learning methods for NN can lead to deadlocks and vague solutions, and it needs a training sample to be prepared. Long training time is the principal disadvantage of NN in robotics, increasing the risk of inappropriate control of expensive equipment, the uniformity of training results for predictions, and the current implementation of NN can only be implemented in a very large-scale integration circuit form.

2.4 *Fuzzy Logic Method*

Zadeh proposed the fuzzy logic as an object with an element membership function in ranges [0, 1] to a set based on the fuzzy set concept [15, 16]. It turns out that fuzzy logic inference can be expressed in the NN form using the membership function to perform the task of neurons and the activation function, considering the neurons' connections as signal transmission. Currently, a lot of fuzzy neural networks roughly explained by the widespread shape of approximators have been developed [17].

2.5 Reinforcement Learning

Reinforcement learning is an approach for handling hybrid optimization problems in machine learning, a structure in which the operator learns how to perform successive decision-making tasks online through interaction with the environment. Reinforcement learning in agent planning is provided by receiving feedback on the outcome of choices made as a reward or punishment without specifying how to achieve the outcome. The reinforcement learning procedure is that the agent first chooses an action from the limited and possible action collection based on observing a situation in the environment and performing that action. Then, the agent receives a predetermined signal from the environment, demonstrating the quality of the operator's action as a reward or punishment. In the next step, the agent transfers to a new environmental status based on the current state [18–20]. In this approach, the agent interacts with the environment by performing a series of actions to find solutions [21, 22]. MDP provides a widely utilized mathematical framework for modeling such problems and consists of four stages [21–23]:

1. A set of states, $S = \{s_1, s_2, ..., s_d\}$
2. A set of actions, $A = \{a_1, a_2, ..., a_m\}$
3. A state transfer function $T = (s'|s, a)$ is a possibility distribution function that a given state s and action into a state s'.
4. A reward function $R = S \times A \times S \to \mathbb{R}$ gives an instant reward when an agent performs an activity and moves from state to state s'.

Using the Markov chain in reinforcement learning, the agent's choice of action is subject to a policy that determines the probability of choosing the action in a specific status. In other words, it determines the effect of the action in an independent state in such a way that the reinforcing learning agent learns to maximize all future rewards [24, 25].

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \quad = \sum_{k=0}^{\infty} \gamma^k . r_{t+k+1}, \tag{2}$$

where t is the time stage and $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$ is the sequence of rewards after the time stage t, and $\gamma \in [0, 1]$ is a deterrent that handles the significance of instant rewards compared to coming rewards and prevents the reward from going to infinity.

One of the best techniques for solving the Markov decision chain problem is the temporal difference technique, which is notable for its good performance, low computational cost, and plain interpretation. The value of a state or action is estimated using the value of other states or actions [26, 27]. Since the proposed technique basis on temporal difference, we express TD as follows:

$$\mathrm{V}(S_t) = \mathrm{V}(S_t) + \alpha\big[r_{t+1} + \gamma \mathrm{V}(S_{t+1}) - \mathrm{V}(S_t)\big] \tag{3}$$

where parameter α is the learning rate that determines how many errors must accept at every step. Parameter γ is the discount rate that characterizes the influence of the following case. The value inside the bracket is a calculation error in the calculation. It calculates the difference between the worth of case $V(S_t)$ and the estimate of the subsequent step and the subsequent reward $r_{t+1}+\gamma \mathrm{V}(S_{t+1})-\mathrm{V}(S_t)$ that the operator tries to minimize this time.

3 Application of Intelligent Approaches in Engineering Control Problems

In this section, we will discuss the applications of intelligent approaches in engineering control problems by reviewing a few works in the literature.

3.1 *Stabilization and Program Control Problems*

Program control and stabilization operations require feedback in the loop. In general, system state vectors are not provided for evaluation, so the available measurement outputs define the control strategy. The robotics and mechatronics standard tasks are similar to speed trajectory tracking and stabilization tasks. These variables can be easily measured at the output of the system. Reference [28] presents a machine learning method for quadcopters. This article presented the approach π_θ using θ as the parameter and is differentiable on parameters. $J(\pi_\theta)$ is the objective function differentiable to θ, for example, the optimization is conducted by the gradient method. For this purpose, probabilistic estimation of the strategy parameters and the mean reward gradient formula is used. The most common method of gradient estimation is formulated as follows:

$$\hat{g} = \widehat{\mathbb{E}_t}\Big[\nabla_\theta \log_{\pi_\theta}(a_t|s_t)\hat{A}_t\Big] \tag{4}$$

where $\widehat{\mathbb{E}_t}$ is the experimental mean for a finite set of instances, $\hat{A}(s_t, a_t) = Q(s_t, a_t) - \hat{V}(s_t)$ represents the advantage function value in time t by changing the sample generation process, and π_θ is the policy enhancement. The dynamic model may be non-differentiable or unknown in this reinforcement learning problem. Thus, the model should be trained, which leads to increasing the gradient estimates variance. For policy optimization in the mentioned article, a solid approach is proposed by incrementally enhancing agent performance. After differentiation of Eq. 4, the objective function is formulated below.

$$J(\theta) = \widehat{\mathbb{E}_{\mathrm{t}}}\Big[min\Big(r(\theta)\hat{A}_t, clip(r(\theta), 1-\varepsilon, 1+\varepsilon)\Big)\Big] \tag{5}$$

where $r(\theta) = \pi_\theta / \pi_{\theta old}$ and ε is the hyperparameter.

By differentiation of Eq. 5, gradient $\hat{g}$ is obtained. The reward function is formulated as follows:

$$r_t\left(e_{xt}, e_{yt}, e_{zt}\right) = \alpha - \sqrt{e_{xt}^2 + e_{yt}^2 + e_{zt}^2} \tag{6}$$

where α is a constant to make sure that each quadcopter is assigned a reward, and e_{xt}, e_{yt}, e_{zt} are the coordinates of the quadcopter.

In reference [29], entropy-based reinforcement learning is considered with a soft membrane vibrating drive for ultra-fast tripod robot gait. Data collection for learning and controller development with feedback are needed for this type of problem. A Gaussian normal distribution policy is defined as the controller: $f_\varphi(s_t) = (\mu_t, \sigma_t)$ in which φ is the controller parameter, σ_t and μ_t refer to the standard deviation and mean, respectively. The action strategies for starting are defined as $N\left(a_t, f_\varphi(s_t)\right)$ and function f_φ is constructed as a neural network. The reward function is presented as follows:

$$r(s_t) = -d_t - \delta\theta_t + c \tag{7}$$

where the root mean square error between the current position of the robot and its final position is shown by d_t, c is the coefficient, and the angular difference between the current and desired position is shown by $\delta\theta_t$. To Maximize Entropy Solution, the optimal solutions policy is formulated as follows:

$$\pi_\alpha^* = \arg\max_\pi \mathbb{E}_{\tau\ P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t\left(\hat{r}(s_t, a_t)\right) + \alpha H(\pi(.|s_t))\right] \tag{8}$$

$$H\left(\pi_\varphi(.|s_t)\right) = \mathbb{E}_{\alpha\sim\pi_\varphi}\left[-log\pi_\varphi(a|s)\right] \tag{9}$$

where α is the entropy temperature in ranges $[0, \infty)$ and $\hat{r}(s_t, a_t) = \mathbb{E}_{\acute{s}\sim P(\pi(\,.|s,a))}\left[r\left(\acute{s}\right)\right]$. The function value should be minimized by stochastic gradient descent.

If we have control goal changing repeatedly, the mentioned reinforcement learning method is not applicable. To solve this problem, you can use a set of state-action-reward, which can be trained to mimic a specific objective in each set. This solution is presented by Puccetti et al. [30] and is tried on a speed control framework.

Bayesian statistical methods are very effective in intelligent systems [31]. A new hypothesis is achieved by recent data from human brain research led to that in specific types of sensorimotor learning, the brain uses Bayesian internal models to optimize performance on specific tasks. The resulting activity of a particular neuronal population can be modeled as a coordinated Bayesian process. The concept of neural signal processing can be utilized in a variety of applications, from rehabilitation engineering to artificial intelligence.

3.2 Controller Tuning

Utilizing fuzzy and adaptive controllers and PID is common in the industry. In adaptive control schemes, both the controller parameters and structure can be changed in response to parameters alteration of the disturbances or controlled object. An overview provides a historical viewpoint on learning methods and adaptive control [7]. In many cases, the structure of the controller is fixed, and only its parameters need to be tuned. It is known how to tune the controller based on a description of the system dynamics. Therefore, it is not easy to obtain in practice, requiring deep system knowledge and potentially open-loop, large-scale measurements. The first proposed algorithm in this area tries to tune a PID controller with the quick reaction of the system model and the sufficiency and cycle of the closed control-loop natural oscillation [32, 33]. Subsequently, an adaptive PID controller and a discriminative adaptive control algorithm were proposed, and the model parameter estimates were used to adjust coefficients [34, 35]. In some cases, especially if the system is unstable, only feedback measurements are possible. The alteration gets to be cumbersome and wasteful in this connection as the operating conditions of the system change. It, therefore, relies on automated methods that can rapidly decide the parameters of the controller without human intercession based on the task. A self-regulating structure starts work in this area.

In reference [36], a multi-parameter self-tuning controller is proposed to control an injection molding machine. The dynamics of a system are explained by the following probabilistic model of discrete time.

$$A\left(q^{-1}\right)y(t) = B\left(q^{-1}\right)u(t-d-1) + C\left(q^{-1}\right)e(t) \tag{10}$$

where an output vector of dimension p is shown by $y(t)$, an input vector of dimension s is indicated by $u(t)$, q^{-1} is the reverse shift operator, $e(t)$ is white Gaussian noise of dimension p, d is the unit time delay, and $(q^{-1}y(t)) = y(t-1)$. The model presented in Eq. 8 is presented by the self-tuning estimation strategy with recursion in k-step as follows:

$$\acute{y}(t+k|t) = \sum_{i=1}^{n_a} \hat{A}_i \hat{y}(t+k-i|t) + \sum_{i=d}^{n_b} \hat{B}_i u(t+k-i) + \sum_{i=d}^{n_c} \hat{C}_i \hat{e}(t+k-i) \tag{11}$$

where $\hat{A}_i$, $\hat{B}_i$, $\hat{C}_i$ indicates the estimated matrices for Eq. 10 and $k = 1, 2, ..., d$. Thus, the optimization problem is reformulated as follows:

$$J = \left\| D_0 \hat{B}_d u(t) + \hat{L}(t) \right\|_{Q1}^2 + \left\| G_0 u(t) + \sum_{i=1}^{r} G_i u(t-1) \right\|_{Q_1}^2 \tag{12}$$

This turns a stochastic optimization problem into a deterministic problem: $\partial J/\partial u(t)$, in which the output of the self-tuning controller indicates by $u(t)$:

$$u(t) = -\left[\left(D_0\hat{B}_d\right)^T Q_1 D_0 \hat{B}_d + G_0^T Q_2 G_0\right]^{-1} \left[\left(d_0\hat{B}_d\right)^T Q_1 \hat{L}(t) + G_0^T Q_2 \sum_{i=1}^{r} G_i u(t-1)\right] \quad (13)$$

The structure adjustment capabilities and additional control of the learning controller must be utilized to fulfill the needs of more complex machines based on the simulation results.

In a study dedicated to self-tuning controllers [37], algorithms were obtained and analyzed by aggregating the least squares estimation (LSE) and tuning the minimum oscillations achieved by the dynamics model. Two theorems are achieved by the primary results assuming convergence of estimating parameters and defining a closed system. Some cross-covariance and output of the output control variable will vanish from the little presumptions of the registry in the first theorem. The second theorem assumes that the control framework may be a common linear likelihood framework of order *n*. When the parameter estimation process is converged, we show that the resulting control law is the control law of most minor variability that can be computed with the known parameters.

3.3 Identification Problems

In reference [38], a method using the bee swarm algorithm to identify linear systems described by differential equations is proposed. To get a model and parameter set that minimizes the prediction error between the model output and the real object, an optimization problem is defined based on the identification problem. The result of the algorithm operation is displayed on the DC motor model.

In reference [39], to adjust the parameters of the PID controller of an evaporator control system while minimizing the system tracking absolute squared error, a heuristic colony competition algorithm was used. The genetic algorithm and Ziegler–Nichols method demonstrate this algorithm's effectiveness.

To determine the lasting magnet synchronous motor model parameters in real time, Rahimi et al. [40] used a heuristic competition algorithm. For this, a minimization process is conducted based on the mean squared error of the system state vector control.

3.4 Optimization Problems

Gradientless search algorithms are widely utilized for all optimization problems due to their versatility. This also applies to NN because NN does not utilize the gradient of the function and does not consider it is differentiable [41]. Their characteristic is that the optimization problem solution is worthy but not ideal. Recently, various biomimetic solutions that borrow ideas from nature are gaining popularity [42, 43]. These include populations [44], swarm and colony algorithms [45–47], evolutionary, etc. A bat algorithm [44] is also known and is related to echolocation-based swarm intelligence. The cuckoo swarm algorithm tunes the PID controller in thyristor series compensation [48] and DC motor control systems [49]. The former was more efficient compared to the Swarm algorithm with the heuristic algorithm.

In reference [50], using support vector algorithms, an optimal control approach is proposed to minimize the bipedal robot's power consumption under a small data sample size and an unknown system dynamics model. The new controller has been integrated into the optimal controller, constraining the robot's joint angles to minimize the energy-related cost function. The energy functional is

$$J_{EE} = \int_0^T \frac{1}{2}\tau^T \tau dt, \tau = g(\Theta) \tag{14}$$

where $g(.)$ is parameterized by NN and Q is a vector of generalized coordinates. The quadratic form support vector machine quality function is

$$J_{SVM} = \min \frac{1}{2}W^T W + \frac{1}{2}C\sum_{i=1}^{N}\xi_i^2 as, \tau_i = w^T\varphi(\Theta_i) + \xi_i \tag{15}$$

where ξ_i is a positive variable, w is a vector of weights, C is a penalty factor, N refers to the training instances number, and $\varphi(.)$ refers to the transformation function of the input space to the input space of higher-order features. The resulting functional includes the aggregation of J_{EE} and J_{SVM}.

In Ref. [51], an improved "learn-learn" search algorithm is utilized by multi-objective optimization of PID controller parameters. This prevents function values from getting stuck in local minima. To this end, there are not only two learners in the learning, but it includes an additional state. Also, parameters ear to inconsistent targets is combined with a blocked device phase where they are blocked. This ensures that each objective cannot collide with another [52]. The results of comparative studies on optimizing the parameters of the PID controller of the DC motor control system utilizing the particle swarm method, the honey bee colony algorithm, and the learning-learning. The last one showed the best results.

3.5 Problems of Iterative Learning

Machine learning is known as one form of artificial intelligence, which is that rather than being explicitly programmed, the systems can be trained by data stored in memory [53]. Based on processing the training data set, a more accurate model is constructed. This allows you to train the model before and on an ongoing basis. The iterative model training process continuously improves the types of relationships between data items, no matter how complex or large. You can continue training in real time using models trained offline.

In Ref. [54], a fault-tolerant control approach is proposed according to the iterative current-loop learning control for recovering the execution of polyphase permanent magnet drives under open-circuit conditions. This method does not need diagnostics and troubleshooting as its main advantage, and torque measurements are sufficient. Iterative learning management, therefore, provides comprehensive knowledge on reliability for modeling uncertainty and the system. We developed a flexible trajectory-assisted control scheme using iterative learning control for a cloud-wheeled robot system to move along a given trajectory and transport cargo simultaneously and performed a system stability analysis [55]. In [56], a human-led iterative learning framework is presented for a trajectory-tracking task in which a controller gets input from the activities of a human agent. Hladowski et al. [57] considered the influence of noise to achieve new results for the dynamic enhancement of iterative learning control laws.

An iterative procedure is presented for planning the milling process in reference [3]. For that, it is necessary to know the machine's technical parameters and the parts' geometrical parameters to form the machine tool trajectory. Tool deviation is a severe problem in which the milling process requires constant review and planning. Dittrich et al. [3] presented the following solution that reduces processing errors by up to 50% by predicting the error between a model of machined shape and actual surface measurements using machine learning methods. Thus, a statistical support vector machine uses the previous process dataset as the training dataset.

4 Conclusions

The modern world trend towards organizations of advanced production types is reflected in intelligent control scientific publications methods in electromechanical systems. Using AI methods, it is possible to solve previously impossible problems of controlling mechatronic systems while at the same time increasing ease of implementation and computational efficiency. The complexities of control tasks for multi-agent systems are inherently non-linear, uncertain, or influenced by external environments and require individual approaches to solving specific problems, for which many tools are proposed. Only by actual experiments, the effectiveness of these learning algorithms on complex systems can be measured. The development of the algorithm itself aims not only to increase the accuracy and speed of learning but also to increase independence from adaptation to various goals and learning strategies that humans strictly set. Developers try to recreate the behavior of living organisms by utilizing natural thoughts in algorithms. Future research establishes a task, usually referred to as "learning for learning," when agents need to select learning strategies and tune meta-parameters.

References

1. Vepa R (1993) Review of techniques for machine learning of real-time control strategies. Intell Syst Eng 2. https://doi.org/10.1049/ise.1993.0009
2. Zaitceva I, Andrievsky B (2022) Methods of intelligent control in mechatronics and robotic engineering: a survey. Electronics (Basel) 11:2443. https://doi.org/10.3390/electronics11152443
3. Dittrich MA, Uhlich F, Denkena B (2019) Self-optimizing tool path generation for 5-axis machining processes. CIRP J Manuf Sci Technol 24. https://doi.org/10.1016/j.cirpj.2018.11.005
4. Gurel S, Akturk MS (2008) Scheduling preventive maintenance on a single CNC machine. Int J Prod Res 46. https://doi.org/10.1080/00207540701487833
5. Mosheiov G (2001) Scheduling problems with a learning effect. Eur J Oper Res 132. https://doi.org/10.1016/S0377-2217(00)00175-2
6. Matni N, Proutiere A, Rantzer A, Tu S (2019) From self-tuning regulators to reinforcement learning and back again. In: Proceedings of the IEEE conference on decision and control. https://doi.org/10.1109/CDC40024.2019.9029916
7. Annaswamy AM, Fradkov AL (2021) A historical perspective of adaptive control and learning. Annu Rev Control 52. https://doi.org/10.1016/j.arcontrol.2021.10.014
8. Fradkov AL (2017) Scientific school of Vladimir Yakubovich in the 20th century. IFAC-PapersOnLine. https://doi.org/10.1016/j.ifacol.2017.08.461
9. Bondarko VA, Yakubovich VA (1992) The method of recursive aim inequalities in adaptive control theory. Int J Adapt Control Signal Process 6. https://doi.org/10.1002/acs.4480060303
10. Gusev SV, Bondarko VA (2020) Notes on Yakubovich's method of recursive objective inequalities and its application in adaptive control and robotics. IFAC-PapersOnLine. https://doi.org/10.1016/j.ifacol.2020.12.1885
11. Lipkovich M (2022) Yakubovich's method of recursive objective inequalities in machine learning. IFAC-PapersOnLine. 55:138–143. https://doi.org/10.1016/j.ifacol.2022.07.301
12. Perel'man II (1991) Analysis of modern adaptive control methods from the stand-point of application to automatization of technological processes. Avtomatika i Telemekhanika
13. Andrievsky BR, Fradkov AL (2021) Speed gradient method and its applications. Autom Remote Control 82. https://doi.org/10.1134/S0005117921090010
14. Abdullah AM, Usmani RSA, Pillai TR, Marjani M, Hashem IAT (2021) An optimized artificial neural network model using genetic algorithm for prediction of traffic emission concentrations. Int J Adv Comput Sci Appl 12. https://doi.org/10.14569/IJACSA.2021.0120693
15. Novák V, Perfilieva I, Močkoř J (1999) Mathematical principles of fuzzy logic.https://doi.org/10.1007/978-1-4615-5217-8
16. Gupta MM, Kiszka JB (2003) Fuzzy sets, fuzzy logic, and fuzzy systems. In: Encyclopedia of physical science and technology. https://doi.org/10.1016/b0-12-227410-5/00270-2
17. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signals Syst 2. https://doi.org/10.1007/BF02551274
18. Mazyavkina N, Sviridov S, Ivanov S, Burnaev E (2021) Reinforcement learning for combinatorial optimization: a survey. Comput Oper Res 134. https://doi.org/10.1016/j.cor.2021.105400

19. Joshi DJ, Kale I, Gandewar S, Korate O, Patwari D, Patil S (2021) Reinforcement learning: a survey. Adv Intell Syst Comput AISC 1311:297–308. https://doi.org/10.1007/978-981-33-4859-2_29
20. Kober J, Bagnell JA, Peters J (2013) Reinforcement learning in robotics: a survey. Int J Robot Res 32:1238–1274. https://doi.org/10.1177/0278364913495721
21. Baird G (2020) Optimising darts strategy using Markov decision processes and reinforcement learning. J Oper Res Soc 71:1020–1037. https://doi.org/10.1080/01605682.2019.1610341
22. Sutton RS, Precup D, Singh S (1999) Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif Intell 112:181–211. https://doi.org/10.1016/S0004-3702(99)00052-1
23. Bäuerle N, Rieder U (2010) Markov decision processes. Jahresber Deutsch Math-Verein 112:217–243. https://doi.org/10.1365/s13291-010-0007-2
24. Kaelbling LP, Littman ML, Moore AW (1996) Reinforcement learning: a survey. Morgan Kaufmann Publishers. https://doi.org/10.1613/jair.301
25. Wang Y, Wang J, Liao H, Chen H (2017) An efficient semi-supervised representatives feature selection algorithm based on information theory. Pattern Recogn 61:511–523. https://doi.org/10.1016/j.patcog.2016.08.011
26. van Seijen H, Mahmood AR, Pilarski PM, Machado MC, Sutton RS (2016) True online temporal-difference learning. J Mach Learn Res 17:1–40
27. Paniri M, Dowlatshahi MB, Nezamabadi-pour H (2021) Ant-TD: ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection. Swarm Evol Comput 64:100892. https://doi.org/10.1016/j.swevo.2021.100892
28. Lopes GC, Ferreira M, da Silva Simoes A, Colombini EL (2018) Intelligent control of a quadrotor with proximal policy optimization reinforcement learning. In: Proceedings—15th Latin American robotics symposium, 6th Brazilian robotics symposium and 9th workshop on robotics in education, LARS/SBR/WRE 2018. https://doi.org/10.1109/LARS/SBR/WRE.2018.00094
29. Kim JI, Hong M, Lee K, Kim D, Park Y-L, Oh S (2020) Learning to walk a tripod mobile robot using nonlinear soft vibration actuators with entropy adaptive reinforcement learning. IEEE Robot Autom Lett 5:2317–2324. https://doi.org/10.1109/LRA.2020.2970945
30. Puccetti L, Kopf F, Rathgeber C, Hohmann S (2020) Speed tracking control using online reinforcement learning in a real car. In: 2020 6th international conference on control, automation and robotics, ICCAR 2020. https://doi.org/10.1109/ICCAR49639.2020.9108051
31. Poon CS (2004) Sensorimotor learning and information processing by Bayesian internal models. In: Annual international conference of the IEEE engineering in medicine and biology—proceedings. https://doi.org/10.1109/iembs.2004.1404245
32. Ziegler JG, Nichols NB (1943) Process lags in automatic control circuits. Trans ASME 65
33. Ziegler JG, Nichols NB (1995) Optimum settings for automatic controllers. InTech 42
34. Astrom KJ, HÄgglund T (2006) Advanced PID control. IEEE Control Syst 26. https://doi.org/10.1109/MCS.2006.1580160
35. Nishikawa Y, Sannomiya N, Ohta T, Tanaka H (1984) A method for auto-tuning of PID control parameters. Automatica 20. https://doi.org/10.1016/0005-1098(84)90047-5
36. Dong CM, Tseng AA (1989) A multivariable self-tuning controller for injection molding machines. Comput Ind 13. https://doi.org/10.1016/0166-3615(89)90042-0
37. Åström KJ, Borisson U, Ljung L, Wittenmark B (1977) Theory and applications of self-tuning regulators. Automatica 13. https://doi.org/10.1016/0005-1098(77)90067-X
38. Erçin O, Çoban R (2012) Identification of linear dynamic systems using the artificial bee colony algorithm. Turk J Electr Eng Comput Sci 20. https://doi.org/10.3906/elk-1012-956
39. Atashpaz Gargari E, Hashemzadeh F, Rajabioun R, Lucas C (2008) Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process. Int J Intell Comput Cybern 1. https://doi.org/10.1108/17563780810893446
40. Rahimi A, Bavafa F, Aghababaei S, Khooban MH, Naghavi SV (2016) The online parameter identification of chaotic behaviour in permanent magnet synchronous motor by Self-Adaptive Learning Bat-inspired algorithm. Int J Electr Power Energy Syst 78. https://doi.org/10.1016/j.ijepes.2015.11.084

41. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80. https://doi.org/10.1007/s11042-020-10139-6
42. Balochian S, Baloochian H (2019) Social mimic optimization algorithm and engineering applications. Expert Syst Appl 134. https://doi.org/10.1016/j.eswa.2019.05.035
43. Kumar S, Kumar A, Shankar G (2018 Crow search algorithm based optimal dynamic performance control of SVC assisted SMIB system. In: 2018 20th national power systems conference, NPSC 2018. https://doi.org/10.1109/NPSC.2018.8771814
44. Sharma P, Sharma K (2022) Fetal state health monitoring using novel Enhanced Binary Bat Algorithm. Comput Electr Eng 101:108035. https://doi.org/10.1016/j.compeleceng.2022.108035
45. Bayati H, Dowlatshahi MB, Hashemi A (2022) MSSL: a memetic-based sparse subspace learning algorithm for multi-label classification. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-022-01616-5
46. Hashemi A, Dowlatshahi MB, Nezamabadi-pour H (2021) Gravitational search algorithm. In: Handbook of AI-based metaheuristics, p 32
47. Hashemi A, Joodaki M, Joodaki NZ, Dowlatshahi MB (2022) Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: a case study in ensemble feature selection. Appl Soft Comput 109046. https://doi.org/10.1016/j.asoc.2022.109046
48. Sethi R, Panda S, Sahoo BP (2015) Cuckoo search algorithm based optimal tuning of PID structured TCSC controller. In: Smart innovation, systems and technologies. https://doi.org/10.1007/978-81-322-2205-7_24
49. Kishnani M, Pareek S, Gupta R (2014) Optimal tuning of PID controller by Cuckoo Search via Lévy flights. In: 2014 international conference on advances in engineering and technology research, ICAETR 2014. https://doi.org/10.1109/ICAETR.2014.7012927
50. Wang L, Liu Z, Chen CLP, Zhang Y, Lee S (2012) Support vector machine based optimal control for minimizing energy consumption of biped walking motions. Int J Precis Eng Manuf 13. https://doi.org/10.1007/s12541-012-0260-7
51. Xiao L, Zhu Q, Li C, Cao Y, Tan Y, Li L (2014) Application of modified teaching-learning algorithm in coordination optimization of TCSC and SVC. Commun Comput Inf Sci. https://doi.org/10.1007/978-3-662-45646-0_5
52. Shouran M, Habil M, Tuning of PID Controller using different optimization algorithms for industrial DC motor. In: 2021 international conference on advance computing and innovative technologies in engineering, ICACITE 2021. https://doi.org/10.1109/ICACITE51222.2021.9404616
53. Judith Hurwitz DK (2018) Machine learning for dummies. IBM Limited Edition
54. Mohammadpour A, Mishra S, Parsa L (2013) Iterative learning control for fault-tolerance in multi-phase permanent-magnet machines. In: 2013 American control conference. IEEE, pp 5929–5934. https://doi.org/10.1109/ACC.2013.6580768
55. Li J, Wang S, Wang J, Li J, Zhao J, Ma L (2021) Iterative learning control for a distributed cloud robot with payload delivery. Assembly Autom 41. https://doi.org/10.1108/AA-11-2020-0179
56. Warrier RB, Devasia S (2016) Iterative learning from novice human demonstrations for output tracking. IEEE Trans Hum Mach Syst 46. https://doi.org/10.1109/THMS.2016.2545243
57. Hladowski L, Galkowski K, Rogers E (2017) Further results on dynamic iterative learning control law design using repetitive process stability theory. In: 2017 10th international workshop on multidimensional (ND) systems, NDS 2017. https://doi.org/10.1109/NDS.2017.8070621