Reinforcement learning for robot research: A comprehensive review and open issues

Tengteng Zhang and Hongwei Mo

Abstract
Applying the learning mechanism of natural living beings to endow intelligent robots with humanoid perception and decision-making wisdom becomes an important force to promote the revolution of science and technology in robot domains. Advances in reinforcement learning (RL) over the past decades have led robotics to be highly automated and intelligent, which ensures safety operation instead of manual work and implementation of more intelligence for many challenging tasks. As an important branch of machine learning, RL can realize sequential decision-making under uncertainties through end-to-end learning and has made a series of significant breakthroughs in robot applications. In this review article, we cover RL algorithms from theoretical background to advanced learning policies in different domains, which accelerate to solving practical problems in robotics. The challenges, open issues, and our thoughts on future research directions of RL are also presented to discover new research areas with the objective to motivate new interest.

Keywords
Reinforcement learning, robotic applications, dexterous manipulation, mobile robotics, deep reinforcement learning, sim-to-real

Introduction
Reinforcement learning (RL), one of the most popular research fields in the context of machine learning, effectively addresses various problems and challenges of artificial intelligence. It has led to a wide range of impressive progress in various domains, such as industrial manufacturing, board games, robot control, and autonomous driving. Robot has become one of the research hot spots all over the world, which is widely used in industry, agriculture, service industry, medical treatment, aerospace, and other fields. A large number of studies on RL algorithms for robots have attracted researchers’ interest and attention. Simultaneously, many well-known research institutes and companies (e.g. DeepMind, UC Berkeley, OpenAI, and Google Brain) have made some achievements in this field but still face enormous challenges.

RL and optimization control theory are studying how to enhance future manipulation of a dynamic system with past data. The goal is to design systems that use richly structured perception, perform planning and control that adequately adapt to environmental changes. Optimal control is to design a controller to maximize the performance of the system in some indicators. The solution of optimal control often uses value function and dynamic programming...
On the basis of Hamilton and Jacobi’s theory, Bellman et al. extended some solutions and gave some solutions with dynamic system state and value function, which is sometimes called optimal return function. According to the Markov decision process (MDP), we can conclude that all optimization problems can be classified as RL problems. In the past decades, some reviews discussed the application of robots, for example, environmental perception, path planning, behavior decision, and motion control. Moreover, significant progress has been made in solving challenging problems across robotic domains using fuzzy control algorithms, genetic algorithm (GA), neural networks (NNs), particle swarm optimization (PSO), ant colony optimization (ACO), and simulated annealing algorithm. The aforementioned studies mainly focus on performance improvement, sampling efficiency, and robot manipulation (i.e. grabbing, handling, and route planning). These methods are often trapped in local optimum and difficult to converge. With the rise of deep learning (DL), it brings a storm to the domain of robot vision. It promotes the rapid development of robot in indoor and outdoor scene recognition, industrial and family services, and multirobot cooperation. The surveys present DL approaches for robot in the literature. Although DL effectively solves some problems of target recognition, grasping positioning, and cooperative learning of robots, it cannot autonomously make decision and control for robotics.

By contrast, the advantages of RL in online adaptability and self-learning features of complex systems for robots have attracted considerable attention. It converges to the optimal control strategy through trial-and-error interacting with environments. There are several typical robot research based on RL (Figure 1). To our knowledge, there is no profound survey specifically discussing RL methods for robot research, regardless of a number of reproducing existing work and interesting results. For convenience, a general taxonomy of the main research works of RL for robotics is shown in Figure 2. The combination of RL and artificial intelligence is of great significance to future studies of robots to solve general artificial intelligence. The list of abbreviations is represented in Table 1.
The rest of this article is organized as follows. The second section introduces the preliminaries of RL. The third section provides a conceptual overview of RL algorithms. The state-of-the-art research of RL for robots are outlined in the fourth section. The fifth section highlights some important challenges and open issues, and offers some perspectives to future research directions. Finally, the sixth section presents concluding remarks.

Preliminaries

Key concepts and terminology

RL is goal-oriented and based on a hypothesis. Robots can learn by trial and error in the process of interaction with the environment based on RL. The ultimate goal is to determine the best sequence of actions to maximize long-term benefits. For connectionist learning, learning algorithms are divided into three types: unsupervised learning, supervised learning, and RL. The characteristic of supervised learning is that the data of learning are labeled. The model is known, that is, we have already told the model what kind of action is correct in what state before learning. In short, we have a special teacher to guide it. It is usually used for regression and classification problems. On the contrary, RL is used to learn without a label but to explore the characteristics of data. It does not directly determine whether a state or action is good or bad but give a reward. Feedback is delayed, data are serialized, and there is a correlation between data and data. The behavior of the agent will affect the subsequent data.

With the exception of the agent and environment, there are eight main elements of an RL system: state $s_t$, action $a_t$, reward $r_t$, policy $\pi$, value function $V^\pi(s_t)$, reward discount factor $\gamma$, state transition probability matrix $P_{st'}$, and exploration rate $\varepsilon$. Based on the basic learning process of RL, as shown in Figure 3, the agent performs actions $a_t$ according to the current strategy at state $s_t$ of time $t$, then arrives at the state $s_{t+1}$ of time $t+1$ and obtains the reward $r_{t+1}$ at time $t$. The observation sequence $H$, states, actions, and reward are
obtained by sampling. The optimal strategy is obtained by value function, which can be used to guide robot manipulation.

Markov decision process

Under uncertain and unstructured environments, MDP\textsuperscript{14} is often used to model decision-making problems. Almost all RL problems can be expressed in the form of MDP. For example, optimal control mainly deals with continuous MDP problems, any part of observable problems can be transformed into MDP problems, and bandits are MDP problems with only one state. Here, bandit is the simplest Markov problem, that is, give you a set of actions and then you choose an action and immediately to get the reward. A typical example of MDP is playing Go (see Figure 4).

RL is a process of running agents through a series of state-action pairs, just as iterative NNs. It extracts information from data by sampling and combines MDP with a large number of state-action pairs. The complex probability distribution model of the reward is associated with it.\textsuperscript{13} A MDP is usually defined as a tuple $\langle S, A, P, R, \gamma \rangle$:

- $S$ is a finite set of states.
- $A$ is a finite set of actions.
- $P$ is state transition probability, that is, the probability matrix of state transition for agent when selecting execution action to next state $P(s'|s, a) = \Pr[S_{t+1} = s'|S_t = s, A_t = a]$ (1)
- $R$ is the reward function $r(s, a, s') = E[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s']$ (2)

The return $G_t$, representing a good or bad state, is the attenuation sum of all rewards from the beginning to the end of sampling for an MDP. The greater the value, the better the state, so as to get more rewards. The cumulative reward value function of agent after state $s_t$ execution strategy $\pi$ is $V_\pi(s_t)$ based on Bellman equation

$$V_\pi(s_t) = E_\pi[R_{t+1} + \gamma V_\pi(s_{t+1})|S_t = s]$$ (5)

In the same way, we can also get the iterative relationship of action-value function

$$Q_\pi(s, a) = E_\pi[R_{t+1} + \gamma Q_\pi(s_{t+1}, A_{t+1})|S_t = s, A_t = a]$$ (6)

Finding an optimal strategy is better to solve the RL problem so that the robot can always gain more than other strategies in the process of interaction with the environment. This problem is transformed into solving the optimal action-value function

$$Q^*(s, a) = \max_\pi Q_\pi(s, a)$$ (7)

Therefore, the optimal strategy can be defined as

$$\pi^*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in A} Q^*(s, a) \\ 0 & \text{else} \end{cases}$$ (8)

Because the Bellman equation\textsuperscript{13} is not linear, nonlinear max function is introduced. Thus, it cannot be solved
directly like Bellman expectation equation to obtain a closed-form solution, which can be iterated by value. It can be solved by value iteration, Q-learning, strategy iteration, or SARSA. When $i \to \infty$, the continuous iteration makes the action-state value function converge, that is, $Q_c \to Q^*_c$. The best action that the agent performs in state $s_t$ is derived as follows

$$a^*_t = \arg \max_{a_t} Q^c(s_t, a_t)$$

### State-of-the-art reinforcement learning algorithms in robotics

Robot research involves many RL algorithms. The generation of training data determines the specific methods used in robot learning. The data needed for robot learning can be generated by the interaction between robot and environment or provided by experts. Then, a modern intelligent robot with autonomous decision-making and learning ability is studied by combining artificial intelligence technology with RL methods. Therefore, value-based RL, policy-based RL, model-based RL, deep reinforcement learning (DRL), meta-RL, and inverse RL (IRL), which have been applied to robots, are reviewed in this section. In addition, Table 2 shows a summary of the strengths and weaknesses of RL methods.

### Value-based reinforcement learning

The value function is the prediction of expectation, accumulation, discount, and future return. Generally, the optimal state- and action-value function $Q^c$ are optimized instead of the state-value function $V^c$ and is updated by $\varepsilon$-greedy policy. The updated policy can be expressed as follows

$$\pi(a|s) = \begin{cases} \frac{\varepsilon}{m} + 1 - \varepsilon & \text{if } a^* = \arg \max_{a \in A} Q^c(s, a) \\ \varepsilon/m & \text{else} \end{cases}$$

DP, Monte Carlo (MC), temporal difference (TD) learning, SARSA, and Q-learning are classical model-free RL algorithms for learning state and action value function. Once the value function is derived, we may get the optimal policy for robot actions.

### Table 2. Summary of the strengths and weaknesses of RL algorithms.

| Category       | Key features                                      | Strengths                                                                 | Weakness                                                                 | References   |
|----------------|---------------------------------------------------|---------------------------------------------------------------------------|--------------------------------------------------------------------------|--------------|
| Value-based RL | Evaluate an action and improve policy, rather than acting directly | Flexible and easy to implement                                           | Not suitable for situations of discontinuous and large state space, difficult to design reward function and consume more memory | 38–49        |
| Policy-based RL| Map a state to an action or to distribute the action | Simpler and easier to converge than value-based RL, directly optimizes the objective function and obtains the optimal strategy | Easy to converge to local optimum and encounter high variance             | 8,50,51–59  |
| Model-based RL | Known model can describe environment and predict the next state and return | Faster training and easy to converge                                       | Difficult to obtain a model and design reward function                  | 32,60–67    |
| DRL            | End-to-end control for raw input image            | Decision making, perception, faster to converge, and lower data association | Data inefficiency, high sample complexity, instability, local optimum, and difficult to design reward function | 68–77        |
| IRL            | No specified reward function                      | Easy to quantify reward function and obtain reward function               | Easy to lead to the same expert policy by different reward function      | 78–85        |
| Meta-RL        | Learn to learn                                    | Flexible, small-scale sample, and faster learning                        | Large-scale parameter space and quadratic gradient                       | 86–98        |

RL: reinforcement learning; DRL: deep reinforcement learning; IRL: inverse reinforcement learning.
There are two main methods of TD learning, that is, on-policy and off-policy, which differ from the way of the Q-value updating. On-policy approach, like SARSA, exploring while learning the optimal strategy, is a model-free online control algorithm of TD. The state space is modeled by a dynamic Bayesian network and updated using a region-based particle filter in the literature. This work makes high-level decisions on the player/stage simulator and the Pioneer robot. The learning process will be smoother and not trapped in a local optimal solution. In addition, the SARSA(λ) based on reverse recognition will be able to effectively learn online for robot, and the data can be discarded after learning. The off-policy methods, like Q-learning, usually update value function with ε-greedy policy but greedy. It directly learns the optimal strategy depending on a series of data generated during training. Thus, it will be greatly affected by sample data and variance of training data, and even affects the convergence of Q function. Tai and Liu adopted a method to explore a corridor environment with the depth information from an RGB-D sensor only, which is used to build such an exploring strategy for robotics by a supervised DL structure and a Q-learning network. Zimmer and Doncieux extract representations dedicated to discrete RL from learning traces generated by neuroevolution results for a faster learning on two simulated robotics tasks. Q-learning and SARSA are recommended for training RL model in a simulated environment and online production environment, respectively.

Policy-based reinforcement learning

In contrast to value-based RL, policy-based RL is to map a state to an action or to distribute the action and then find the best mapping relationship by strategic optimization. The policy search methods mainly include random policy search and deterministic policy search. Assuming that the expectation of initial state harvest is taken as the optimization objective, the following formula represents the gradient of the derivation of θ

\[ \nabla \mathcal{J}(\theta) = E_{\pi_{\theta}}[\nabla \log \pi_{\theta}(s, a) Q_{\lambda}(s, a)] \]  

(14)

where \( \nabla \log \pi_{\theta}(s, a) \) is the score function and the parameter \( \theta \) of policy function is updating in the direction of \( \theta + \alpha \nabla \log \pi_{\theta}(s, a) V(s) \). Sutton et al. designed the following policy function (15) based on softmax policy

\[ \pi_{\theta}(s, a) = \frac{e^{\phi(s,a)^T \theta}}{\sum_b e^{\phi(s,b)^T \theta}} \]  

(15)

and the score function was calculated by Gauss policy (16).

Silver et al. presented a framework for DPG algorithms to ensure adequate exploration and learn a deterministic target policy from an exploratory behavior policy in high-dimensional action spaces. UCB proposed trust region policy optimization (TRPO) to effectively optimize large nonlinear policies such as NNs. The TRPO algorithm outperforms prior methods on a range of challenging policy learning tasks, for example, learning simulated robotic swimming, hopping, and walking gaits. DeepMind used the idea of deep Q-network (DQN) extended from Q-learning algorithm to modify their deterministic strategy gradient method and proposed a deep deterministic strategy gradient algorithm (DDPG) based on actor-critic (AC) framework. In the simulation environment MuJoCo, the target of the robot grasping operation in continuous action space is realized. The robust model-free approach attacks the limitation of a large number of training episodes to find solutions for robotics dexterous manipulation and legged locomotion.

Mnih et al. found that asynchronous advantage AC algorithm can adapt to both discrete and continuous space. Levine et al. trained complex manipulation skills for a PR2 robot end-to-end with guided policy search. Policy gradient with Q-learning significantly outperformed AC and Q-learning on Atari games testing. Additionally, path consistency learning minimized a notion of soft consistency error along multistep action sequences extracted from both on-policy and off-policy traces. Haarnoja et al. derived a soft Q-learning algorithm by applying deep energy-based policies to maximum entropy policies so that skills can be transferred between tasks for simulated swimming and walking robots. In a recent study, the generalization of the learned policy is successfully verified on physical robots in rich and complex environments using policy-gradient-based method. In addition, a state-of-the-art survey focuses on leveraging prior knowledge on the policy structure and creating data-driven surrogate models of the expected reward to find effective policy search algorithms. Therefore, although policy-based RL usually
converges to local optimum and encounters high variance, it directly optimizes the objective function and obtains the optimal strategy.

**Model-based reinforcement learning**

The value-based and policy-based RL is model free, which learns directly from value function and policy function. The state \( s \) and action \( a \) are used as input to predict the next state \( s' \), that is, state transition probability model \( P(s'_{t+1}|s_t, a_t) \), and the reward prediction model \( R_{t+1} = R(R_{t+1}|s_t, a_t) \) (if forecasting the reward of the environment). Synthesized from past experience, model-based RL aims to learn a model that predicts future observations and addresses these shortcomings of lacking the behavioral flexibility constitutive of general intelligence by endowing agents with a model of the world (see Figure 5).

Sutton integrated a Dyna architecture for learning, planning, and reacting based on approximating DP.\(^{60} \) In contrast to Dyna, the Dyna-2 architecture\(^{61} \) is designed, which separates the experience of interacting with the environment and model predictions. A unified framework that ranges from model-based to model-free methods was designed for learning the continuous control policies by backpropagation.\(^ {62} \) A variety of challenging, underactuated, physical control problems are solved, including reaching, grabbing, and tracking of a robot arm. Model-based methods improved model-free RL for continuous control tasks of simulated robot using Q-learning with experience replay and effectively accelerated learning by Gu et al.\(^{63} \) Polydoros and Nalpantidis reviewed the applications of model-based RL for robotics and outlined the state-of-the-art in both algorithms and hardware.\(^ {32} \) Unlike classical model-based RL and planning methods, the model-based RL and model-free RL were combined to interpret predictions from a dynamic model to construct implicit plans in arbitrary ways by NN architecture.\(^ {64,65} \) Since the state transition models need to be known in model-free RL, the deficiencies of the learned models have limited the utility for robot learning and planning. An advanced study on discrete-action domains using TreeQ and ATreeC has been presented by Farquhar et al.,\(^ {67} \) which is training end-to-end and shows the benefit of a box-pushing domain and a set of Atari games over previous approaches. Remarkably, in a study,\(^ {66} \) authors trained temporal difference models to train with model-free learning and that was used for model-based control on a range of robot continuous control tasks, for example, reaching target locations (real-world Sawyer robot), pushing a puck to a random target, and training the cheetah to run at target velocities.

**Deep reinforcement learning**

Deep learning,\(^ {27} \) another branch of machine learning, usually consists of multilayer nonlinear operation units. Regarding the output of the lower layer as input, the deep abstract feature representation is automatically acquired from a large number of training data. Significant successes have been achieved in image processing, speech recognition, natural language processing, robot control, and other domains.\(^ {10,68,69,71} \) Compared to traditional multilayer NN algorithms, DL is conducive to alleviating the gradient dispersion and local optimum and eliminating the curse of dimension caused by high dimensional data. The representative structures for DL include as a deep belief network, stacked autoencoder, recurrent neural network, and convolutional neural network (CNN). RL permits agents or robots to learn decision making by millions of interactions with the environment across a variety of different domains. Consequently, as an artificial intelligence method closer to human thinking, various DRL combining the perceptive ability of DL with the decision-making ability of RL achieves direct control from raw input to output by end-to-end learning.

Mnih et al. pioneered a DQN algorithm that combined CNN with traditional Q-learning for approaching Q-learning method with nonlinear functions.\(^ {72} \) There are three main aspects for DQN to improve traditional Q-learning based on experience replay mechanism: (1) approximating the value function with deep CNN; (2) training the learning process of RL by experience replay; and (3) setting up the target network independently to deal with TD error. Figure 6(a) shows the DQN architecture and the training process is given by Figure 6(b), and the detailed learning processes are as follows

\[
L(\theta) = E_{s,a} \sim \rho(\cdot) [\text{Target}Q - Q(s, a; \theta)]^2
\]  
\[
\text{Target}Q = E_{s' \sim S} [r + \gamma \max_{a'} Q(s', a'; \theta')] s, a
\]
\[
\n\nabla_{\theta} L(\theta) = \mathbb{E}_{s,a \sim \rho(\cdot),r \sim \mathcal{S}[\theta]} [r \circ \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta) \nabla Q(s, a; \theta)]
\]

where \(L(\theta)\) and Target\(Q\) are the loss function and target function, respectively. \(\rho(\cdot)\) denotes the probability distribution of selecting action \(a\) in a given environment \(s\). At iteration time \(t+1\), the network weight parameter \(\nabla_{\theta} L(\theta)\) is updated by two identical networks, that is, Main\(Net\) and Target\(Net\). In the literature,\(^{73}\) the performance comparison among DQN algorithms was summarized.

Inverse reinforcement learning

Most multistep decision-making problems are difficult to obtain reward functions in complex environments. IRL\(^{78}\) that has made a breakthrough in the field of robotics reversely solves reward function in MDP based on the hypothesis of optimal expert decision trajectories.\(^{79-81}\) Robot is able to learn how to make complex decisions while the reward function is not specified.

When the dimension of state space is very large and high dimension, the classical IRL methods have been proven exceptionally not effective enough. The states and actions in the model are replaced by DNN,\(^{83}\) which has achieved significant performance in large and complex systems. Similarly, another comprehensive study of a practical and scalable IRL algorithm, that is, adversarial inverse reinforcement learning (AIRL), was presented in Fu et al.\(^{84}\) This approach is able to recover robust reward functions and policies under variation in the underlying domain. Peng et al.\(^{85}\) applied an adaptive stochastic regularization method for adversarial learning to AIRL to yield variational AIRL algorithm, which tends to recover smoother reward functions that is closer to the ground truth reward. More fluent interactions between human and robot were effectively performed for specified human–robot collaboration task based on co-operative IRL.\(^{70}\) A robot successfully learned to set a table according to a demonstrator’s preferences by Brown et al.\(^{103}\) Experimentally, a recent
study improves navigation performance for human-aware robot in a limited visual.

**Meta-reinforcement learning**

Generally, a good reward function is often difficult to design, and a large number of training samples are needed to make the performance of the model a certain height. To reduce the excessive dependence on big data and realize small sample learning, Berkeley Blog published an article called learning to learn (i.e. meta-learning). The goal of meta-learning (MTL) is to learn and quickly train an adaptive model to a new task from a series of learning tasks. Some recent studies mainly focus on model-based MTL, metric-based MTL, and gradient descent-based learning.

Romera-Paredes et al. presented zero-shot learning approaches with the aim of addressing automatic classification problems. The researchers from OpenAI obtained a general system by one-shot imitation learning, which turned any demonstrations into robust policies. Another study on few-shot learning (FSL) from the definition to core issues was proposed by Wang et al. FSL, mimics human, combines prior knowledge with few supervised experience to rapidly generalize to a new task. In a recent study, the latest development of MTL is briefly introduced and developed an off-policy meta-RL algorithm to improve sample efficiency and the effectiveness in sparse reward problems. Hence, advances in MTL have led to great breakthroughs and a flurry of research to robotic systems.

**Reinforcement learning for various robotic applications**

The above-outlined techniques provide an operable method for robotics technology. By means of prior knowledge, behavior discretization, approximation of value function, mental rehearsal, and preconstructed strategies, the robots have the abilities of decision-making and self-learning. The state-of-the-art research to robot in recent years, as reported in the literature, is summarized in Table 3.

| App.                          | Value-based RL | Policy-based RL | Model-based RL | DRL | IRL | Meta-RL |
|-------------------------------|----------------|----------------|----------------|-----|-----|---------|
| Dexterous manipulation        | 42,45,107,108  | 54,99,109–112  | 113,114        | 10,54,101,114–117 | 70,118 | 97,119  |
| Trajectory and route tracking | 120–123        | 36             | 124,125        | 125–127 | 36   | 86      |
| Navigation                    | 48,128,129     | 130            | 131            | 48,129,132–139 | 80,140 | 141,142 |
| Path planning                 | 49,143–146     | 114,147        | 125            | 118,127,139,148–150 | 151–153 | 86      |

RL: reinforcement learning.

**Dexterous manipulation**

At present, the dexterous manipulation of robot is a huge challenge in continuous motion space with respect to complex environments. Using model-free and pure vision-based RL method to perform high-sensitivity manipulator was first proposed by Katya et al., which got grid of the dependence on precise models of robot, for example, kinematics model, dynamics model, interaction force, high fidelity tactile sensor, or joint position sensor. The robot successfully controls a pneumatic five-fingered hand rotating object in a Gazebo simulation environment. The Hindsight Experience Replay technique that can be combined with an arbitrary off-policy RL algorithm was exploited by Andrychowicz et al. to achieve the manipulation tasks. Similarly, various types of robot manipulation problems were solved very efficiently even though in cluttered environments, for example, shifting, sliding, and pick-and-place. Normally, a certain amount of training data is needed to be sampled to update policies in every iteration step of policy optimization, and the cost of acquiring training data is higher in real robotic scenarios. A low-priced and highly flexible multifingered RIO hand with multiple operational skills was successfully developed, which can rotate valve, push abacus, grab objects, and so on. Later, the asynchronous normalized advantage functions with safety constraints represented one of the few algorithms that are capable of alleviating the human intervention for complex 3D manipulation tasks in simulation environment and training on real physical robots. To learn more complex operation skills, faster learning and less interaction with the environment are the key to real-world applications by a handful of trials. Rajeswaran et al. only utilized a small number of artificial demonstrations to perform relocation and door opening (see Figure 7(b)). Using distributed RL to learn operating skills in the simulator by Andrychowicz et al. to achieve an unprecedented level of dexterity on a five-fingered manipulator without relying on human demonstrations (see Figure 7(c)). A recent study improved sampling efficiency and learning stability with deep P network (DPN) and double DPN. The two-arm cooperative robot successfully flips the handkerchief and folds the T-shirt with a limited number of samples (see Figure 7(a)). Li Zhang and Mo
et al. solved various maneuvering dynamics and uncertain external disturbances of the humanoid-like mobile manipulator based on the advanced online kinematics redundancy solution of the neural dynamic optimization algorithm.\textsuperscript{112}

Despite the powerful capability of an individual robot, there are significant breakthroughs in the field of multirobot object manipulation and robot-assisted surgery.\textsuperscript{155} Excitingly, a recent study significantly improves performance for human–robot collaboration.\textsuperscript{107}

	extbf{Trajectory and route tracking}

To realize dynamic obstacle avoidance, the robot tracks the reference route in a certain error range according to a specific certain control law and finally reaches the reference point of a preset geometric route in a partially observable nonlinear dynamic environment. Therefore, it is critical to improve the real time and adaptability of obstacle avoidance and navigation by trajectory and route tracking techniques. Figure 8 shows the sketch map of robotic trajectory and route tracking. Conventional algorithms are easy to fall into local optimum, oscillate in similar obstacle groups, swing in narrow channels, and even cannot identify the path. Additionally, target reference points are not reachable, which eventually leads to errors and instability in tracking and dynamic obstacle avoidance.

AC compensators were designed,\textsuperscript{120} which was used to reduce tracking error of a multiple DOF industrial robot manipulator. A variety of real-world robotic manipulation tasks, such as dish placement and pouring, used policy optimization to adaptively sample trajectories and effectively to learn good global costs for complex robotic motion skills from user demonstrations.\textsuperscript{36} Compared to traditional PID control, the improved DRL algorithm was employed to effectively handle the control problem of trajectory tracking for autonomous underwater vehicle.\textsuperscript{126} A study by Nagabandi et al. presented a hybrid algorithm that training NN dynamic models along with a small number of samples was able to accelerate learning and follow arbitrary trajectories.\textsuperscript{124} A DDPG model was trained to track the optimal route towards large-scale outdoor applications.\textsuperscript{125} Long et al. presented a safe and efficient collision avoidance method using a mixture of learned agents and potential fields.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure7.png}
\caption{Dexterous manipulation: (a) T-shirt folding by NEXTAGE,\textsuperscript{117} (b) manipulator operating system of humanoid mobile robot,\textsuperscript{112} (c) OpenAI: a five-fingered humanoid hand trained with reinforcement learning manipulated a block from an initial configuration to a goal.\textsuperscript{111}}
\end{figure}
Perceptual information is to build the map for the robot. In the past decades, all sorts of algorithms that require camera, radars, and other sensors are developed for robot navigation and the adaptability of traditional methods are both difficult to navigate when some special signs or specific environmental characteristics are unknown. The RL methods that search an optimal or suboptimal path from the start point to the goal point enable mobile robots to self-explore and self-learn by interacting with the environment. Huang et al. improved Q-learning algorithm to reduce the probability of collisions under dynamic environments. A method of end-to-end training with mapless motion planner was employed for mobile robots in unseen virtual and real environments without any prior demonstrations. It would be better if the model is more generalizable when transferring to unseen environments, thus, a hybrid RL model was presented to solve a real-world vision-language navigation task. The target-driven robot navigation technique was used to memorize valuable points’ information about the environment and generalize to a real robot scenario with a model trained in simulation. Another study presented an end-to-end differentiable neural architecture to successfully navigate along paths not encountered. To acquire the multifaceted navigation skills, Chen et al. mapped height-map image observations to motor commands of wheel-legged robot, which significantly improved the quality of obstacle avoidance. Service robot that is capable of autonomous long-range navigation and motion greatly enhances the ability of transporting goods, medicines, luggage, and so on. Assume that the NN architecture and the process of RL search reward are combined with motion planning control algorithm, the robot can navigate in a long range. Hao-Tien et al. employed AutoRL to automatically search for the best feedback and network architecture by means of large-scale hyper-parameter optimization and to learn path-following navigation behaviors. This method better generalizes to new environments, though it has sampling inefficiency. Similarly, the robot learned point-to-point navigation policies end-to-end. The authors designed the probabilistic roadmaps for sampling-based path planning to enable long-range navigation. After training, it can adapt to a variety of different environments. Combining probabilistic roadmaps and AutoRL instead of manual adjusted RL local planner successfully completed long-range indoor navigation. To better verify the effectiveness of the algorithm, the physical platform was utilized to verify navigation and obstacle avoidance in complex scenarios. Experimentally, the robustness, controllability, and precision of robots have been fully studied in practical applications.

**Navigation**

Fast and robust autonomous navigation under various scenarios by means of environmental perception and location techniques becomes a major topic in researchers. Generally, in all these applications, the robot complete navigation on the basis of the sketch map can be seen in Figure 9. Over the past decades, all sorts of algorithms that require cameras, radars, and other sensors are developed for robot to detect obstacles in the navigation environment. And the perceptual information is to build the map for the robot to plan a path around obstacles. Currently, the representative method is the simultaneous localization and mapping technology that builds maps incrementally by estimating the moving positions. However, the calculation and the adaptability of traditional methods are both difficult to navigate when some special signs or specific environmental characteristics are unknown.

![Figure 8](image-url) The sketch map of robotic route and trajectory tracking: (a) Route tracking: $d$ is the error distance between robot and reference route. (b) Trajectory tracking: For arbitrary robot pose $q = [x, y, \theta]^T$, the agent enables to track reference pose $\hat{p}_t = [x_t, y_t, \theta_t]^T$ and velocity $[v, w]_t$, where $v$ and $w$ are linear and angular velocities of an agent.
target using TRPO to optimize NN end-to-end. Another similar study on human–robot interaction focused on developing natural social navigation behavior algorithms,\textsuperscript{140} which enabled collision avoidance, leader–follower, and split-and-rejoin based on expert demonstration. A visual navigation control method made up of low-level behaviors and a metalevel policy was presented for three different simulated robots to avoid obstacles in new compound environments with both learn and sequence robot behaviors.\textsuperscript{141} In a recent study,\textsuperscript{142} the authors hold a self-adaptive visual navigation method based on meta-RL to learn and adapt novel scenes.

**Path planning**

The changeability and complexity of the robot motion environment put forward higher requirements for dynamic obstacle avoidance and path planning. The purpose of path planning is to plan a collision-free optimal or sub-optimal path from the starting point to the target point in a given space, and to be as smooth and safe as possible. Path planning mainly consists of global path planning based on a known model environment and local path planning on the basis of unknown sensor environment. Although traditional algorithms have obtained a series of achievements,\textsuperscript{24,159,163,164} there are still many shortcomings in accuracy, stability, and robustness. The environment model of robotic path planning is shown in Figure 10.

Generally, it is difficult to address the problems of mobile robot path planning in dynamic scene by classical methods. Jaradat et al. applied Q-learning algorithm to limit the number of states and successfully reached its target without collision.\textsuperscript{143} Later, many improved Q-learning algorithms were presented to save storage and decrease the searching scope. The proposed method reduced the energy consumption and time complexity instead of repeatedly updating Q-table by Konar et al.\textsuperscript{144} Experimentally, the $\varepsilon$-greedy exploration and Boltzmann exploration were used to shrink orientation angle and path length under heuristic searching strategies by Li et al.\textsuperscript{145} Roy et al. utilized image processing techniques and RL methods to plan the shortest path for mobile robots.\textsuperscript{146} Another similar study,\textsuperscript{147} in which robot followed the observed demonstration trajectories by visual servo tracking control, tended to design a new robot demonstration learning framework with image-based planning method. To avoid the policy degradation caused by the method based on the value function\textsuperscript{147} was to optimize the strategy with parameters by the idea of gradient rising and maximizing the cumulative expected reward.

To find feasible collision-free and time-efficient paths, the authors hold a decentralized multiagent collision avoidance algorithm that encoded the estimated time and searched for a collision-free velocity vector by Chen et al.\textsuperscript{118} Wu et al. robustly overcame the problems of robot local trajectory planning based on data-driven representation learning.\textsuperscript{148} Similar to previous studies, the perceptive ability of convolutional NNs and end-to-end learning mode of RL is critical to robot path planning. For the instability of the robot training stage and the sparsity of the environment state space, updating NN and increasing the greedy rule probability enhanced the ability of local planning by inputting lidar signal and local target position.\textsuperscript{149} The raw sensor measurements were directly mapped to robot commands to find time-efficient and collision-free paths for multirobot systems by Long et al.\textsuperscript{127} Thus, the applicability and scalability were experimentally demonstrated in a large-scale scenario with 100 robots. Later, Francis et al. proposed a sampling-based robot path planning algorithm combining DRL with long-range motion planning methods for different navigation tasks.\textsuperscript{139}

On a real robotic wheelchair platform, Kim et al. attempted to adopt three-layer architecture for socially adaptive path planning.\textsuperscript{151} A large number of demonstration trajectories generated by experts are utilized to infer the cost function and then plan an optimal path for robot in various dynamic environments. For planetary rovers, a recent study developed a soft value iteration network,\textsuperscript{152} which represented policy with the action probability distribution and effectively trained gradients based on IRL. Wu et al. successfully proposed a policy for online trajectory planning for free-floating space robot without dynamic and kinematic models.\textsuperscript{166} Recent practical experiments\textsuperscript{150,153,167,168} make a series of huge breakthroughs for mobile robot, sake-like robot, cleaning, and maintenance robot. These great achievements of real-world greatly promote the development of robotic applications in the future.
Demonstration and imitation learning

Traditional RL algorithms are generally high computational cost, high complexity, time consuming, and poor scalability for policy acquisition. Imitation learning, similar to supervised learning, is an available way for transferring movement skills from a human expert demonstrations to the robot. Although model-free RL is widely used, the supervised information provided by human experts can promote robot learning to imitate the next skill. Robots quickly determine strategies for new scenarios if they learn from their own peer experts (i.e. through teleoperation or demonstrations) or human expert demonstrations, and even generalize models in the underlying assignment of tasks, such as learning to manipulate new objects by watching a video. Moreover, demonstration learning can avoid directly modeling environment and reduce the complexity of robot action programming.

To reduce the system interaction time for approaching, grasping, and picking up complex objects, Duan et al. proposed one-shot imitation learning. The authors hold a neural net trained with pairs of demonstrations, where input the first demonstration and a state sampled from the second demonstration on a family of block stacking experiments. The meta-imitation learning, unlike the prior one-shot imitation, inclined to learn new manipulation tasks end-to-end from a single visual demonstration in complex unstructured environments without learning each skill from scratch. Another similar research built up prior knowledge through MTL. APR2 arm and a Sawyer arm successfully learn to place, push, and pick-and-place new objects combining prior knowledge with a single video of human manipulation demonstration.

Pfeiffer et al. firstly employed NN to learn a target-oriented end-to-end navigation model, which directly learned from the demonstration for motion planning of autonomous ground robots. It is difficult to design a scripted motion planner or controller in previous work. Therefore, only a small amount of demonstration data was leveraged to train end-to-end visuomotor policies with large visual and dynamics variations for robot manipulation tasks by Zhu et al. In response to the challenges of disaster relief, constrained personnel and equipment, a system of learning from minimal human demonstration was built to fast perform actions and learn to mimic navigation behaviors. Additionally, Nguyen et al. developed a general framework of imitation learning with indirect intervention based on visual navigation and language assistance to search for objects in photorealistic indoor environments.

For dexterous multifingered hands, the pretrain policies with behavior cloning were derived based on demonstration argument policy gradients. To tackle the problems of achieving compound and multistage tasks without providing any direct supervision, Yu et al. presented a method for earning and composing convolutional NN policies. It is better to prevent policies from deviating from human demonstrations and guide the exploration of the manipulator with trajectory tracking assistant reward. In a recent study, the human demonstration used for imitation learning provided an intuitive way to evaluate state representation methods for robot hand–eye coordination learning in both state dimension reduction and controllability. A variety of order fulfillment and kitchen serving tasks were successfully learned in the context of decomposing a human demonstration into primitives at metatest stage (see Figure 11). The recent advances of novel robot tasks via imitation learning from demonstration were reviewed in a survey. The updated taxonomy and classification of current methods are helpful for future research both in theory and in practice.

Sim-to-real

Running, climbing, falling, and climbing are inherent instincts of human beings. To our knowledge, the performance of robots has been unsatisfactory with respect to walking gracefully or grasping naturally. The coordination of gait movement and dexterity of robotic manipulator has always been a difficult problem in the industry. Over the
past decades, it is easier to trap in seemingly smaller obstacles in the physical world than in simulation. The universal simulation environments are given in Table 4. Almost none of these unpredictable obstacles, that is, surface friction, structural flexibility, vibration, sensor delay, and poor actuator transformation of the robot itself, and so on, can be assumed in advance by mathematical models. With the development of technology, the gap between simulation and reality is gradually bridged.

James et al. proposed a simple and highly scalable approach to compute robot trajectories in simulator and successfully achieve end-to-end manipulation and control in the real world.\textsuperscript{176} To avoid tedious manual tuning or calibration, Tan et al. and Bharadhwaj et al. did a good job of porting what the policies learned from the simulator and off-policy data to the real robot.\textsuperscript{33,177} The simulation environment provides a basic platform for the analysis, synthesis and offline programming of robot system with high real-time, versatility and authenticity, and low cost. The simulation models of various tasks in virtue of the large amount of randomized simulation data help us enhance the real-world data, which speeds up the process of robot learning and training. How to transfer simulation to reality is still critical and challenging for robotics. Recently, Liu et al. obtain appealing performance by transferring the trained policy in simulation environments to the real-world scenarios, which significantly reduce training costs and improve the generalization capability for robot control tasks.\textsuperscript{178}

**Challenges, open issues, and directions**

**Real-world challenges and open issues**

Overall, RL methods have recently made notable progress for robotic application due to supercomputational power, frontier algorithms, and large-scale dataset. However, sample inefficiency, higher training costs, uncertain models, dimensional disaster, and so on, have restricted the development of RL in robotic domain. Furthermore, a large amount of RL research of robots are still at the stage of simulation, which are far from performing well for real-world problems. There still a long way to go for robot to learn and master various skills that human does in a shorter time. In general, the challenges and open issues needed to be solved for future research are as follows:

- In research on robot, data play an important role in decision making and the evaluation of learning. The scale of action and state space increase exponentially with the increase of the number of features for RL tasks, which leads to dimension disaster. More data and computation will be needed when exploring states and actions.
- Exploration and exploitation face a central issue: either exploitation gives more knowledge about the environment to make the best decision or exploring

![Figure 11](image-url). One-shot hierarchical imitation learning of compound visuomotor tasks. (Left): training robot by learning primitive behaviors from human demonstrations. (Right): testing the skills of performing compound tasks by PR2 robot.\textsuperscript{35}

**Table 4.** The simulation environments.

| Platform     | Application                                      |
|--------------|--------------------------------------------------|
| MuJoCo       | Robot physical simulator                         |
| OpenAI Gym   | Diverse scenarios (e.g. robot control, Go, Cart-Pole) |
| DeepMind Lab | 3D labyrinth scene (robot)                       |
| TORCS        | Racing car simulator                             |
| SIGVerse     | Robot simulator (e.g. dynamics, perception, communication) |
| AI2-THOR     | Simulation environment similar to real-world scenes |
The robot optimization based on RL is extremely complex and nonconvex optimization problems. It is designed under convex assumption and then applied to nonconvex objective function. These nonconvex optimization algorithms are still lemmas or extensions of convex optimization (convex analysis). The breakthroughs in algorithmic theory of these nonconvex optimization problems are generally attributed to finding the “convex” structure and sometimes impossible to solve the optimal control problem.

The NN is a kind of nonlinear approximation. The disadvantage of nonlinear approximation is the existence of local minimum and the difficulty of optimization. In the process of RL, the data are generated by the interaction between agents and environment, and the adjacent data are not independent and identically distributed. It is not stable to use the data directly to train the NN.

Due to the influence of obstacles in workspace, complex coupling characteristics, and nonlinearity, the trajectory planning and control of robot are faced with many problems. The challenges at the interface of nonlinear coupling and learning must be solved before we can build robust, safe robot learning systems that interact with an uncertain physical environment.

Future research directions

1. Maximum entropy DRL: We expect RL methods to show good performance for robot decision making and control in the real world. However, complex sample, the higher dimension of data space, and poor convergence restrict the development of general artificial intelligence. Therefore, it is necessary to optimize the hyperparameters, which limits the applicability of the robot for an unstructured and complex environment. Maximum entropy deep reinforcement learning (MEDRL) provides a basis for constructing hierarchical strategies that can solve complex and sparse reward tasks through probabilistic reasoning while eliminating the trouble of adjusting hyperparameters. It is used for improving the search strategy and preventing convergence to a local optimum. Compared with the deterministic policy search method, the MEDRL has stronger consistency and robustness. Learning expressive energy policy from soft Q-learning and combining nonstrategic updates with soft AC is to maximize expected returns and entropy in random situations.

2. Semantics to operations: Designing reward functions and exploring time are the obstacles for robot to applying RL methods to the real world. Previous research on robotic learning skills require manually preset the reward functions, which are then applied to be optimized. Although robots cannot understand tasks by observing or depending on human language, researchers have helped robots understand semantic concepts and complete tasks by combining a small amount of annotated data with RL methods. Learning to use experience to understand events of human demonstrations, imitating and learning human actions and understanding semantic categories (e.g. toys and pens) have become a prevalent trend in the future development for robots.

3. Shared system of clouds robot: Although the research of RL has made great progress for robots, such as grasping, stacking, navigation, and so on, the kinds of behaviors that robot master are limited. In addition, it takes a long time for a single robot to collect sufficient training data sets. Recording these actions, by which robots iteratively improve the network for evaluating different states and action values through RL, are expected for robots while exploring different ways to accomplish a task. Learning motor skills and intrinsic physical models directly from experience enable each robot to obtain a copy of the updated network before performing the next stage of action. For robots with different locations and configurations in the real world, the shared system of clouds robot appears to be an efficient approach for collecting large
amounts of data in a short time, accelerating robot learning and even constructing highly generalized representations of individual robot.

4. **Bio-inspired learning:** The computation and energy consumption generated by training robot increases exponentially, which is not sustainable development. For example, the maximum entropy optimization and RL algorithms are utilized to successfully predict the metabolite concentration of erythrospora. Additionally, the ACO and PSO have solved the intelligent optimization problem of multiagent cluster. There are also many key technologies in artificial intelligence, such as artificial neural network, artificial immune system, GA, and so on, which come from the study of biological science. In the future, bio-inspired learning based on RL is a topic worthy of study, it will provide new ideas and technical means to solve the problems of robot applications.

**Conclusions**

Over the past few decades, robots have been unable to achieve high intelligence due to the constraints of algorithms and hardware. This article presents a comprehensive survey on various kinds of RL algorithms and models to robot research. We first give a tutorial of RL from fundamental concepts to advanced methods and highlight their advantages in addressing the challenges brought about by robot research. Subsequently, this article discusses the state-of-the-art robot research on the basis of RL, for example, dexterous manipulation, navigation, trajectory and route tracking, path planning, demonstration and imitation learning and sim-to-real. Despite this article has laid a solid foundation and opened up new research interests to robots, there remain many different factors affecting reproducibility of RL algorithms for real-world robot tasks. Finally, the existing challenges, open issues as well as important future research directions are highlighted to push the important research forward.

**Acknowledgments**

This work was supported in part by my tutor and lab classmates. We thank the authors of Figures 1, 7, and 9 for authorizing us to use their pictures in this article.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

**ORCID iD**

Tengteng Zhang https://orcid.org/0000-0002-8116-3252

**References**

1. Sutton RS and Barto AG. *Reinforcement learning: an introduction.* Cambridge, MA: MIT Press, 2018.
2. Mahadevan S and Theocharous G. Optimizing production manufacturing using reinforcement learning. In: *FLAIRS conference*, Menlo Park, CA, 18 May 1998, pp. 372–377. AAAI Press.
3. Silver D, Hubert T, Schrittwieser J, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* 2018; 362(6419): 1140–1144.
4. Kober J, Bagnell JA, and Peters J. Reinforcement learning in robotics: a survey. *Int J Robot Res* 2013; 32(11): 1238–1274.
5. Isele D, Rahimi R, Cosgun A, et al. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 21–25 May 2018, pp. 2034–2039. IEEE.
6. Bahrin MAK, Othman MF, Azli NN, et al. Industry 4.0: a review on industrial automation and robotic. *J Technol* 2016; 78(6–13): 137–143.
7. Heess H, Dhruva TB, Sriram S, et al. Emergence of locomotion behaviours in rich environments. CoRR 2017; abs/1707.02286. Available at: https://arxiv.org/pdf/1707.02286.
8. Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies. *J Mach Learn Res* 2016; 17(1): 1334–1373.
9. Al-Shedivat M, Bansal T, Burda Y, et al. Continuous adaptation via meta-learning in nonstationary and competitive environments. CoRR 2017; abs/1710.03641. Available at: https://arxiv.org/pdf/1710.03641.
10. Levine S, Pastor P, Krizhevsky A, et al. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int J Robot Res* 2018; 37(4-5): 421–436.
11. Recht B. A tour of reinforcement learning: the view from continuous control. *Ann Rev Control Robot Autonom Syst* 2019; 6: 253–279.
12. Ke L and Jitendra M. Learning to optimize. In: *5th international conference on learning representations*, Toulon, France, 24–26 April 2017.
13. Bellman R. On the theory of dynamic programming. *Proc Natl Acad Sci USA* 1952; 38(8): 716–719.
14. Otterlo MV and Wiering M. Reinforcement learning and Markov decision processes. *Reinf Learn* 2012; 12: 3–42.
15. Chen S, Li Y, and Kwok NM. Active vision in robotic systems: a survey of recent developments. *Int J Robot Res* 2011; 30(11): 1343–1377.
16. Pol RS and Murugan M. A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods. In: *International conference on industrial
instrumentation and control (ICIC), Pune, India, 28–30 May 2015, pp. 1339–1344. IEEE.
17. Foukarakis M, Leonidis A, Antona M, et al. Combining finite state machine and decision-making tools for adaptable robot behavior. In: International conference on universal access in human-computer interaction, Heraklion, Crete, Greece, 22–27 June 2014, pp. 625–635. Springer.
18. Precup RE and Hellendoorn H. A survey on industrial applications of fuzzy control. Comput Ind 2011; 62(3): 213–226.
19. Boubaker O. The inverted pendulum benchmark in nonlinear control theory: a survey. Int J Adv Rob Syst 2013; 10(233): 1–9.
20. Sciavicco L and Siciliano B. Modelling and control of robot manipulators. Berlin/Heidelberg: Springer Science & Business Media, 2012.
21. Bingil Z and Karahan O. A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Expert Syst Appl 2011; 38(1): 1017–1031.
22. Karami AH and Hasanzadeh M. An adaptive genetic algorithm for robot motion planning in 2D complex environments. Comput Electr Eng 2015; 43: 317–329.
23. He W, Chen Y, and Yin Z. Adaptive neural network control of an uncertain robot with full-state constraints. IEEE Trans Cybern 2015; 46(3): 620–629.
24. Zhang Y, Gong DW, and Zhang JH. Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing 2013; 103: 172–185.
25. Liu J, Yang J, Liu H, et al. An improved ant colony algorithm for robot path planning. Soft Comput 2017; 21(19): 5829–5839.
26. Miao H and Tian YC, Dynamic robot path planning using an enhanced simulated annealing approach. Appl Math Comput 2013; 222: 420–437.
27. LeCun Y, Bengio Y, and Hinton G. Deep learning. Nature 2015; 521(7553): 436–444.
28. Lenz I, Honglak L, and Saxena A. Deep learning for detecting robotic grasps. Int J Robot Res 2015; 34(4–5): 705–724.
29. Zhu H, Gupta A, Rajesvaran A, et al. Dexterous manipulation with deep reinforcement learning: efficient, general, and low-cost. In: IEEE international conference on robotics and automation (ICRA), Montreal, QC, Canada, 20–24 May 2019, pp. 3651–3657. IEEE.
30. Cui Y, Matsubara T, and Sugimoto K. Pneumatic artificial muscle-driven robot control using local update reinforcement learning. Adv Robot 2017; 31(8): 397–412.
31. Xie Z, Clary P, Dao J, et al. Iterative reinforcement learning based design of dynamic locomotion skills for Cassie. CoRR 2019; abs/1903.09537. Available at: https://arxiv.org/pdf/1903.09537.
32. Polydoros AS and Nalpantidis L. Survey of model-based reinforcement learning applications on robotics. J Intell Robot Syst 2017; 86(2): 153–173.
33. Tan J, Zhang T, Coumans E, et al. Sim-to-real: learning agile locomotion for quadruped robots. In: 14th Conference on Robotics - Science and Systems, Pittsburgh, Pennsylvania, USA, 26–30 June 2018. DOI: 10.15607/RSS.2018.XIV.010.
34. Nagabandi A, Yang G, Asmar T, et al. Learning image-conditioned dynamics models for control of under-actuated legged milirobots. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 1–5 October 2018, pp. 4606–4613. IEEE.
35. Kalashnikov D, Irpan A, Pastor P, et al. QT-opt: scalable deep reinforcement learning for vision-based robotic manipulation. CoRR 2018; abs/1806.10293. Available at: https://arxiv.org/pdf/1806.10293.
36. Finn C, Levine S, and Abbeel P. Guided cost learning: deep inverse optimal control via policy optimization. In: International conference on machine learning, New York City, NY, USA, 19–24 June 2016, pp. 49–58.
37. Watkins CJCH and Dayan P. Q-learning. Mach Learn 1992; 8(3–4): 279–292.
38. Rust J. Dynamic programming. Mineola, NY: Dover Publications, 2003.
39. Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 1970: 57(1): 97–109.
40. Peidro A, Reinoso O, Gil A, et al. An improved Monte Carlo method based on Gaussian growth to calculate the workspace of robots. Eng Appl Artif Intell 2017; 64: 197–207.
41. Yu H, Mahmood AR, and Sutton RS. On generalized bellman equations and temporal-difference learning. J Mach Learn Res 2018; 19(1): 1864–1912.
42. Huang X, NaghdY F, Du H, et al. Temporal difference (TD) based critic-actor adaptive control for a fine hand motion rehabilitation robot. In: Billingsley J and Brett P (eds) Mechatronics and machine vision in practice 3, Cham: Springer, 2018, pp. 195–207.
43. Martin J, Wang J, and Englot B. Sparse gaussian process temporal difference learning for marine robot navigation. In: 2nd Conference on Robot Learning, PMLR, Zürich, Switzerland, 29–31 October 2018, pp. 179–189.
44. Szepesvári C. Algorithms for reinforcement learning. Synth Lect Artif Intell Mach Learn 2010; 4(1): 1–103.
45. Ramachandran D and Gupta R. Smoothed sarsa: reinforcement learning for robot delivery tasks. In: IEEE international conference on robotics and automation, Kobe, Japan, 12–17 May 2009, pp. 2125–2132. IEEE.
46. Harutyunyan A, Bellemare MG, Stepleton T, et al. Q(τ) with off-policy corrections. In: International conference on algorithmic learning theory, Bari, Italy, 19–21 October 2016, pp. 305–320. Springer.
47. Precup D, Sutton RS, and Dagsupta S. Off-policy temporal-difference learning with function approximation. In: International conference on machine learning, San Francisco, CA, 28 June 2001, pp. 417–424.
48. Tai L and Liu M. A robot exploration strategy based on Q-learning network. In: IEEE international conference on real-time computing and robotics (RCAR), Angkor Wat, Cambodia, 6–10 June 2016, pp. 57–62. IEEE.
49. Zimmer M and Doncieux S. Bootstrapping Q-learning for robotics from neuro-evolution results. IEEE Trans Cogn Dev Syst 2017; 10(1): 102–119.
50. Fan T, Long P, Liu W, et al. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int J Robot Res* 2020; 39(7): 856–892.

51. Sutton RS, McAllester D, Singh S, et al. Policy gradient methods for reinforcement learning with function approximation. In: *Advances in neural information processing systems*, Cambridge, MA: MIT Press, 2000, pp. 1057–1063.

52. Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization. In: *International conference on machine learning*, Lille, France, 6–11 July 2015, pp. 1889–1897. New York, NY: ACM.

53. Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: *International conference on machine learning*, Beijing, China, 27 January 2014.

54. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous control with deep reinforcement learning. 2015.

55. Mnih V, Badia AP, Mirza M, et al. Asynchronous methods for deep reinforcement learning. In: *International conference on machine learning*, New York, NY, USA, 19–24 June 2016, pp. 1928–1937. IEEE.

56. Nair A, McGrew B, Andrychowicz M, et al. Overcoming exploration in reinforcement learning with demonstrations. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 21–25 May 2018, pp. 6292–6299. IEEE.

57. O’Donoghue B, Munos R, Kavukcuoglu K, et al. Combining policy gradient and Q-learning. *CoRR* 2016; abs/1611.01626. Available at: https://arxiv.org/pdf/1611.01626.

58. Nachum O, Norouzi M, Xu K, et al. Bridging the gap between value and policy based reinforcement learning. In: *Advances in neural information processing systems 30 (NIPS 2017)*, 2017, pp. 2775–2785. New York, NY: ACM.

59. Haarnoja T, Tang H, Abbeel P, et al. Reinforcement learning with deep energy-based policies. In: *Proceedings of the 34th international conference on machine learning*, Sydney, NSW, Australia, 6–11 August 2017, pp. 1352–1361.

60. Sutton RS. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: *Machine learning proceedings 1990*. Austin, Texas, 21–23 June 1990, pp. 216–224. Elsevier.

61. Silver D, Richard RS, and Müller M. Sample-based learning and search with permanent and transient memories. In: *Proceedings of the 25th international conference on machine learning*, Helsinki, Finland, July 2008, pp. 968–975. New York, NY: ACM.

62. Heess N, Wayne G, Silver D, et al. Learning continuous control policies by stochastic value gradients. In: *Advances in neural information processing systems 28 (NIPS 2015)*, 2015, pp. 2944–2952. New York, NY: ACM.

63. Gu S, Lillicrap T, Sutskever I, et al. Continuous deep q-learning with model-based acceleration. In: *International conference on machine learning*, New York, NY, USA, 11 June 2016, pp. 2829–2838. ACM.

64. Oh J, Singh S, and Lee H. Value prediction network. In: *Advances in neural information processing systems 30 (NIPS 2017)*, 2017, pp. 6118–6128. New York, NY: ACM.

65. Racanière S, Weber T, Reichert D, et al. Imagination-augmented agents for deep reinforcement learning. In: *Proceedings of the 31st international conference on neural information processing systems*, Siem Reap, Cambodia, 14–18 October 2018, pp. 5694–5705. ACM.

66. Pong P, Gu S, Dalal M, et al. Temporal difference models: model-free deep RL for model-based control. In: *6th international conference on learning representations (ICLR 2018)*, Vancouver, Canada, 30 April–3 May 2018. OpenReview.net.

67. Farquhar G, Rocktäschel T, Igl M, et al. TreeQN and AtreeC: differentiable tree-structured models for deep reinforcement learning. 2017.

68. Amodei D, Ananthanarayanan S, Anubhai R, et al. Deep speech 2: end-to-end speech recognition in English and Mandarin. In: *International conference on machine learning*, New York, NY, USA, 11 June 2016, pp. 173–182. ACM.

69. Young T, Hazarika D, Poria S, et al. Recent trends in deep learning based natural language processing. *IEEE Comput Intel Mag* 2018; 13(3): 55–75.

70. Malik D, Palaniappan M, Fisc FJ, et al. An efficient, generalized bellman update for cooperative inverse reinforcement learning. In: *International Conference on Machine Learning*, Stockholmsmässan, Stockholm SWEDEN, 10–15 July 2018, pp. 3394–3402.

71. Madani A, Aarnaout R, Mofrad M, et al. Fast and accurate view classification of echocardiograms using deep learning. *NPJ Digit Med* 2018; 1(1): 1–6.

72. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature* 2015; 518(7540): 529–533.

73. Luong NC, Hoang DT, Gong S, et al. Applications of deep reinforcement learning in communications and networking: a survey. *IEEE Commun Surv Tutor* 2019; 21(4): 3133–3174.

74. Hasselt HV, Guez A, and Silver D. Deep reinforcement learning with double q-learning. In: *Proceedings of the 30th AAAI conference on artificial intelligence*, Phoenix, Arizona USA, 12–17 February 2016, pp. 2094–2100. AAAI Press.

75. Schaul T, Quan J, Antonoglou I, et al. Silver, prioritized experience replay. *CoRR* 2015; abs/1511.05952. Available at: https://arxiv.org/pdf/1511.05952.

76. Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning. In: *Proceedings of the 33rd international conference on machine learning*, New York City, NY, USA, 19–24 June 2016, pp. 1995–2003.

77. Hausknecht M and Stone P. Deep recurrent q-learning for partially observable MDPs. In: *AAAI fall symposium on sequential decision making for intelligent agents (AAAI-SDMIA15)*, Arlington, Virginia, USA, 2015.

78. Ng AY and Russell SJ. Algorithms for inverse reinforcement learning. In: *International conference on machine learning*, Stanford, CA, USA, 29 June 2000, pp. 2–9.

79. Krishnan S, Garg A, Liaw R, et al. SWIRL: a sequential windowed inverse reinforcement learning algorithm for robot tasks with delayed rewards. *Int J Robot Res* 2019; 38(2–3): 126–145.
References

80. Kretzschmar H, Spies M, Sprunk C, et al. Socially compliant mobile robot navigation via inverse reinforcement learning. *Int J Robot Res* 2016; 35(11): 1289–1307.

81. Vasquez D, Okal B, and Arras KO. Inverse reinforcement learning algorithms and features for robot navigation in crowds: an experimental comparison. In: *IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, USA, 14–18 September 2014, pp. 1341–1346. IEEE.

82. Sun S, Zhao X, Li Q, et al. Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision. *Adv Robot* 2020; 34(13): 887–901.

83. Wulfmeier M, Ondruska P, and Posner I. Maximum entropy deep inverse reinforcement learning. *CoRR* 2015; abs/1507.04888. Available at: https://arxiv.org/pdf/1507.04888.

84. Fu J, Luo K, and Levine S. Learning robust rewards with adversarial inverse reinforcement learning. In: *6th international conference on learning representations (ICLR 2018)*, Vancouver, Canada, 30 April–3 May 2018. Openreview.net.

85. Peng XB, Kanazawa A, Toyer S, et al. Variational discriminator bottleneck: improving imitation learning, inverse RL, and gans by constraining information flow. In: *6th international conference on learning representations (ICLR 2018)*, Vancouver, Canada, 30 April–3 May 2018. Openreview.net.

86. Houthooft R, Chen Y, Isola P, et al. Sample-efficient deep reinforcement learning with imaginary rollouts. In: *Advances in neural information processing systems 31 (NIPS2018)*, 2018, pp. 5400–5409.

87. Wang JX, Kurth-Nelson Z, Tirumala D, et al. Learning to reinforcement learn. *CoRR* 2016; abs/1611.05763. Available at: https://arxiv.org/pdf/1611.05763.

88. Wang JX, Kurth-Nelson Z, Kumaran D, et al. Prefrontal cortex as a meta-reinforcement learning system. *Nat Neurosci* 2018; 21(6): 860–868.

89. Mishra N, Rohaninejad M, Chen X, et al. A simple neural attentive meta-learner. *CoRR* 2017; abs/1707.03141. Available at: https://arxiv.org/pdf/1707.03141.

90. Finn C, Abbeel P, and Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: *Proceedings of the 34th international conference on machine learning*, Sydney, NSW, Australia, 6–11 August 2017, pp. 1126–1135. ACM.

91. Romera-Paredes B and Torr P. An embarrassingly simple approach to zero-shot learning. In: *International symposium on machine learning*, Lille, France, 1 June 2015, pp. 2152–2161. Springer.

92. Duan Y, Andrychowicz M, Stadie B, et al. One-shot imitation learning. In: *Advances in neural information processing systems 30 (NIPS2017)*. Long Beach, CA, USA, 4–9 December 2017, pp. 1087–1098.

93. Wang Y, Yao Q, Kwok J, et al. Few-shot learning: a survey. *CoRR* 2019; abs/1904.05046v1. Available at: https://arxiv.org/pdf/1904.05046v1.

94. Rakelly K, Zhou A, Finn C, et al. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In: *Proceedings of the 36th international conference on machine learning*, Long Beach, CA, 24 May 2019, pp. 5331–5340.

95. Duan Y. Meta learning for control. Dissertation UC Berkeley, 2017.

96. Finn C, Yu T, Zhang T, et al. One-shot visual imitation learning via meta-learning. In: *Proceedings of the 1st annual conference on robot learning*, California, USA, 13–15 November 2017, pp. 357–368.

97. Nagabandi A, Clavera I, Liu S, et al. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In: *2nd workshop on meta-learning at NeurIPS 2018*, Montréal, Canada, 2–8 December 2018.

98. Humphri J, Galashov A, Hasenelever L, et al. Meta reinforcement learning as task inference. *CoRR* 2019; abs/1905.06424. Available at: https://arxiv.org/pdf/1905.06424.

99. Chatzilygeroudis K, Vassiliades V, Stulp F, et al. A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Trans Robot* 2020; 36(99): 328–347.

100. Hester T, Vecerik M, Pietquin Q, et al. Deep Q-learning from demonstrations. In: *Proceedings of the 32th AAAI conference on artificial intelligence*, New Orleans, Louisiana, USA, 2–7 February 2018, pp. 3223–3230. AAAI Press.

101. Zhao W, Queralta JP, Qingqing L, et al. Towards closing the sim-to-real gap in collaborative multi-robot deep reinforcement learning. In: *International conference on robotics and automation engineering*, Singapore, Singapore, 20–22 November 2020, pp. 1–6. IEEE.

102. Thabet M, Patacchiola M, and Cangelosi A. Sample-efficient deep reinforcement learning with imaginary rollouts for human-robot interaction. 2019.

103. Brown DS, Cui Y, and Niekum S. Risk-aware active inverse reinforcement learning. In: *Proceedings of the 2nd conference on robot learning*, Zürich, Switzerland, 29–31 October 2018, pp. 362–372.

104. Finn C, Xu K, and Levine S. Probabilistic model-agnostic meta-learning. In: *Advances in neural information processing systems 31 (NIPS2018)*, 2018, pp. 9516–9527. New York, NY: ACM.

105. Zou F and Feng J. Hierarchical meta learning. *CoRR* 2019; abs/1904.09081. Available at: https://arxiv.org/pdf/1904.09081.

106. Choi J, Lee L, Park J, et al. Zero-shot learning and knowledge transfer in music classification and tagging. In: *Machine learning for music discovery workshop*, The 36th international conference on machine learning (ICML), Long Beach, California, USA, 9–15 June 2019.

107. Roveda L, Maskani J, Franceschi P, et al. Model-based reinforcement learning variable impedance control for human-robot collaboration. *J Intell Robot Syst* 2020; 100: 417–433.

108. Andrychowicz M, Wolski F, Ray A, et al. Hindsight experience replay. In: *Advances in neural information processing systems 30 (NIPS 2017)*, 2017, pp. 5048–5058. New York, NY: ACM.

109. Berscheid L, Meißner P, and Kroeger T. Robot learning of shifting objects for grasping in cluttered environments. In: *2019 IEEE/RSJ international conference on intelligent systems 30(NIPS 2017)*, Long Beach, CA, USA, 4–9 December 2017, pp. 1087–1098.
115. Katya KD1, Staley EW, Johannes MS, et al. In-hand robotic manipulation for a soft robotic hand from human demonstrations. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Daejeon, Korea (South), 9–14 October 2016, pp. 3786–3793. IEEE.

116. Andrychowicz M, Baker B, Chociej M, et al. Learning dexterous in-hand manipulation. Int J Robot Res 2020; 39(1): 3–20.

117. Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. CoRR 2017; abs/1709.10087. Available at: https://arxiv.org/pdf/1709.10087.

118. Chen YF, Liu M, Everett M, et al. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017, pp. 3389–3396. IEEE.

119. Katya KD1, Staley EW, Johannes MS, et al. In-hand robotic manipulation via deep reinforcement learning. In: International conference on neural information processing systems 30 (NIPS 2016), Barcelona, Spain, December 2016, pp. 1–5.

120. Rajeswaran A, Kumar V, Gupta A, et al. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. CoRR 2017; abs/1709.10087. Available at: https://arxiv.org/pdf/1709.10087.

121. Yang Y, Caluwaerts K, Iscen A, et al. Data efficient reinforcement learning for legged robots. In: Proceedings of the conference on robot learning, Osaka, Japan, 12 May 2020, pp. 1–10.

122. Ota K, Jha DK, Oiki T, et al. Trajectory optimization for unknown constrained systems using reinforcement learning. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS), Macau, China, 3–8 November 2019, pp. 3487–3494. IEEE.

123. Xiao J, Li L, Zou Y, et al. Reinforcement learning for robotic time-optimal path tracking using prior knowledge. CoRR 2019; abs/1907.00388. Available at: https://arxiv.org/ftp/arxiv/papers/1907/1907.00388.

124. Nagabandi A, Kahn G, Fearing RS, et al. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018, pp. 7559–7566. IEEE.

125. Wei M, Wang S, Zheng J, et al. UGV navigation optimization aided by reinforcement learning-based path tracking. IEEE Access 2018; 6: 57814–57825.

126. Yu R, Shi Z, Huang C, et al. Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle. In: IEEE 36th chinese control conference (CCC), Dalian, China, 26–28 July 2017, pp. 4958–4965. IEEE.

127. Long P, Fanl T, Liao X, et al. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In: IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017, pp. 6252–6259. IEEE.

128. Huang L, Qu H, Fu M, et al. Reinforcement learning for mobile robot obstacle avoidance under dynamic environments. In: Pacific rim international conference on artificial intelligence, Nanjing, China, 28–31 August 2018, pp. 441–453. Springer.

129. Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017, pp. 3357–3364. IEEE.

130. Li MM, Jiang R, Ge SS, et al. Role playing learning for socially concomitant mobile robot navigation. CAAI Trans Intell Technol 2018; 3(1): 49–58.

131. Wang X, Xiong W, Wang H, et al. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In: Proceedings of the european conference on computer vision (ECCV), Munich, Germany, 8–14 September, 2018, pp. 37–53. Springer.

132. Wang D, Deng H, and Pan Z. MRCDDL: multi-robot coordination with deep reinforcement learning. Neurocomputing 2020; 406: 68–76.

133. Tai L, Paolo G, and Liu M. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation. In: IEEE/RSJ international conference on intelligent robots and systems (IROS), Vancouver, BC, Canada, 24–28 September 2017, pp. 31–36. IEEE.

134. Nguyen DV, Vuong TL, Kieu HD, et al. Vision memory for target object navigation using deep reinforcement learning: an empirical study. In: IEEE international conference on systems, man, and cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018 pp. 3267–3273. IEEE.

135. Shah P, Fiser M, Faust A, et al. FollowNet: robot navigation by following natural language directions with deep reinforcement learning. In: Proceedings of the conference on robot learning, Osaka, Japan, 12 May 2020, pp. 1–10.
reinforcement learning. *CoRR* 2018; abs/1805.06150. Available at: https://arxiv.org/pdf/1805.06150.

136. Chen X, Ghadirzadeh A, Folkesson J, et al. Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Madrid, Spain, 1–5 October 2018, pp. 3110–3116. IEEE.

137. Chiang HTL, Faust A, Fiser M, et al. Learning navigation behaviors end-to-end with autoRL. *IEEE Robot Autom Lett* 2019; 4(2): 2007–2014.

138. Faust A, Oslund K, Ramirez O, et al. PRM-RL: long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. In: *IEEE international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 21–25 May 2018, pp. 5113–5120. IEEE.

139. Francis A, Faust A, Chiang HT, et al. Long-range indoor navigation with PRM-RL. *IEEE Trans Robot* 2020; 36(4): 1115–1134.

140. Fahad M, Chen Z, and Guo Y. Learning how pedestrians navigate: a deep inverse reinforcement learning approach. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Madrid, Spain, 1–5 October 2018, pp. 819–826. IEEE.

141. Salman H, Singhal P, Shank T, et al. Learning to sequence robot behaviors for visual navigation. *CoRR* 2018; abs/1803.01446. Available at: https://arxiv.org/pdf/1803.01446.

142. Wortsman M, Ehsani K, Rastegari M, et al. Learning to sequence robotic navigation tasks by combining reinforcement learning and sampling-based planning. In: *IEEE/RSJ international conference on robotics and automation (ICRA)*, Brisbane, QLD, Australia, 21–25 May 2018, pp. 5113–5120. IEEE.

143. Zhang and Mo 21

144. Konar A, Chakraborty IG, Singh SJ, et al. A deterministic reward-free policy iteration algorithm for reinforcement learning based on Q-learning. In: *IEEE international conference on robotics and automation (ICRA)*, Paris, France, 31 May–31 August 2020, pp. 8440–8446. IEEE.

145. Roveda L, Maskani J, Franceschi P, et al. Learning to walk a tripod mobile robot using nonlinear soft vibration actuators with entropy adaptive reinforcement learning. *J Intell Robot Syst* 2020; 92(2): 2317–2324.

146. Lit A, Xu X, and Zuo L. Dynamic path planning of a mobile robot with improved Q-learning algorithm. In: *IEEE international conference on information and communication, Lijiang, China, 8–10 August 2015*, pp. 409–414. IEEE.

147. Roy N, Chattopadhay ray R, Mukherjee A, et al. Implementation of image processing and reinforcement learning in path planning of mobile robots. *Int J Eng Sci* 2017; 7(10): 15211–15213.

148. Hazara M and Kyrki V. Reinforcement learning for improving imitated in-contact skills. In: *IEEE international conference on humanoid robots*, Cancun, Mexico, 15–17 November 2016, pp. 194–201. IEEE.

149. Lei X, Zhang Z, and Dong P. Dynamic path planning of unknown environment based on deep reinforcement learning. *J Robot* 2018; 2018: 1–10.

150. Wang Y, Fang Y, Lou P, et al. Deep reinforcement learning based path planning for mobile robot in unknown environment. *J Phys Conf Ser* 2020; 1576: 012009.

151. Kim J. Socially adaptive path planning in human environments using inverse reinforcement learning. *Int J Soc Robot* 2016; v8(1): 51–66.

152. Pfueger M, Agha A, and Sukhatme GS. Rover-IRL: inverse reinforcement learning with soft value iteration networks for planetary rover path planning. *IEEE Robot Autom Lett* 2019; 4(2): 1387–1394.

153. Bing Z, Lemke C, Cheng L, et al. Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning. *Neural Netw* 2020; 129: 323–333.

154. Johannink T, Bahl S, Nair A, et al. Residual reinforcement learning for robot control. In: *International conference on robotics and automation (ICRA)*, Montreal, QC, Canada, 20–24 May 2019, pp. 6023–6029. IEEE.

155. Gao X, Jin Y, Dou Q, et al. Automatic gesture recognition in robot-assisted surgery with reinforcement learning and tree search. In: *2020 IEEE international conference on robotics and automation (ICRA)*, Paris, France, 31 May–31 August 2020, pp. 8440–8446. IEEE.

156. Garcia MAP, Montiel O, Castillo O, et al. Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation. In: Melin P, Castillo O, Ramirez EG, Kacprzyk J, and Pedrycz W (eds) *Analysis and design of intelligent systems using soft computing techniques*, Berlin, Heidelberg: Springer, 2007; Vol. 41, pp. 790–798.

157. Manikas TW, Ashenayi K, and Wainwright RL. Genetic algorithms for autonomous robot navigation. *IEEE Instrum Meas Mag* 2007; 10(6): 26–31.

158. Garcia MAP, Montiel O, Castillo O, et al. Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation. In: Melin P, Castillo O, Ramirez EG, Kacprzyk J, and Pedrycz W (eds) *Analysis and design of intelligent systems using soft computing techniques*, Berlin, Heidelberg: Springer, 2007; Vol. 41, pp. 790–798.

159. Garcia MAP, Montiel O, Castillo O, et al. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl Soft Comput* 2009; 9(3): 1102–1110.

160. Jiang CF and Chang YC. Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments. *IEEE Trans Fuzzy Syst* 2011; 19(2): 379–392.

161. Ahmadzadeh S and Ghanavati M. Navigation of mobile robots in complex environments using inverse reinforcement learning. *IEEE Trans Robot* 2016; 32(6): 1309–1332.
163. Tennety S, Sarkar S, Hall EL, et al. Support vector machines
based mobile robot path planning in an unknown environment. In: ASME
dynamic systems and control conference, Hollywood, California, USA, 12–14 October, 2009, pp. 395–401. ASME.
164. Ismail AT, Sheta A, and Al-Weshah M. A mobile robot path
planning using genetic algorithm in static environment. J Comput Sci 2008; 4(4): 341–344.
165. Vakanski A, Janabi-Sharifi F, and Mantegh I. An image-
based trajectory planning approach for robust robot programming by demonstration. Robot Autonom Syst 2017; 98: 241–257.
166. Wu YH, Yu ZC, Li CY, et al. Reinforcement learning in
dual-arm trajectory planning for a free-floating space robot. Aerosp Sci Technol 2020; 98: 105657.
167. Lakshmanan AK, Mohan RE, Ramalingam B, et al. Complete
coverage path planning using reinforcement learning for Tetromino based cleaning and maintenance robot. Autom Construct 2020; 112: 103078.
168. Wang B, Liu Z, Li Q, et al. Mobile robot path planning in
dynamic environments through globally guided reinforcement
learning. IEEE Robot Autom Lett 2020; 5(4): 6932–6939.
169. Pfeiffer M, Schaeuble M, Nieto J, et al. From perception to
decision: a data-driven approach to end-to-end motion planning for autonomous ground robots. In: IEEE international conference on robotics and automation (ICRA), Singapore, 29 May–3 June 2017, pp. 1527–1533. IEEE.
170. Zhu Y, Wang Z, Merel J, et al. Reinforcement and imitation
learning for diverse visuomotor skills. In: 6th international conference on learning representations (ICLR 2018), Vancouver, Canada, 30 April–3 May 2018. Openreview.net.
171. Wigness M, Rogers JG, and Navarro-Serment LE. Robot
navigation from human demonstration: learning control behaviors. In: IEEE international conference on robotics and automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018, pp. 1150–1157.
172. Nguyen K, Dey D, Brockett C, et al. Vision-based navigation
with language-based assistance via imitation learning with indirect intervention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Long Beach, CA, USA, 15–20 June 2019, pp. 12527–12537. IEEE.
173. Yu T, Abbeel P, Levine S, et al. One-shot hierarchical imitation
learning of compound visuomotor tasks. CoRR 2018; abs/1810.11043. Available at: https://arxiv.org/pdf/1810.11043.
174. Jin J, Dehghan M, Petrich L, et al. Evaluation of state rep-
resentation methods in robot hand-eye coordination learning from demonstration. CoRR 2019; abs/1903.00634. Available at: https://arxiv.org/pdf/1903.00634.
175. Ravichandar H, Polydoros AS, Chernova S, et al. Recent
advances in robot learning from demonstration. Ann Rev 2020; 3: 297–330.
176. James S, Davison AJ, Johns E, et al. Transferring end-to-end
visuomotor control from simulation to real world for a multi-stage task. In: Proceedings of the 1st annual conference on robot learning, California, USA, 13–15 November 2017, pp. 334–343.
177. Bharadhwaj H, Wang Z, Bengio Y, et al. A data-efficient
framework for training and sim-to-real transfer of navigation policies. In: IEEE international conference on robotics and automation (ICRA), Montreal, QC, Canada, 20–24 May 2019, pp. 782–788. IEEE.
178. Liu N, Cai Y, Lu T, et al. Real-sim-real transfer for real-
world robot control policy learning with deep reinforcement
learning. Appl Sci 2020; 10(5): 1555.
179. Aza NA, Shahmansoorian A, and Davoudi M. From inverse
optimal control to inverse reinforcement learning: a historical
review. Ann Rev Control 2020; 50: 119–138.
180. Haarnoja T. Acquiring diverse robot skills via maximum
entropy deep reinforcement learning. Dissertation UC Berkeley, 2018.
181. Deisenroth MP, Neumann G, Peters J, et al. A survey on
policy search for robotics. Found Trends Robot 2013; 2(1–2): 1–142.
182. Ding G, Chen M, Zhao S, et al. Neural image caption generation with weighted training and reference. Cogn Comput 2018; 11: 763–777.
183. Liao Y, Wu J, and Luo X. A formal model for robot to
understand common concepts. In: Intelligent computing-proceedings of the computing conference. London, 16 July 2019, pp. 517–526. Springer.
184. Liu B, Wang L, Liu M, et al. Lifelong federated reinforcement
learning: a learning architecture for navigation in cloud robotic systems. IEEE Robot Autom Lett 2019; 4(4): 4555–4562.
185. Cannon WR, Zucker JD, Baxter DJ, et al. Prediction of
metabolite concentrations, rate constants and post-translational regulation using maximum entropy-based simulations with application to central metabolism of neu-
rospora Crassa. Processes 2018; 6(6): 63.
186. Soltoggio A, Stanley KO, and Risi S. Born to learn: the inspira-
tion, progress, and future of evolved plastic artificial neural networks. Neural Netw 2018; 108: 48–67.