International Journal on Emerging Technologies (Special Issue on ICRIET-2016) **7**(2): 76-80(2016)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Reinforcement Learning in Robotics

***Vipin Vibhute*[*], *Dr. Yogesh Angal*[**] *and Dr. B. Suryakanth*[***]**

[*]*D.Y. Patil College of Engineering,Akurdi, Pune, Maharashtra, India*
[**]*JSPM's Bhivarabai Sawant Institute of Technology & Research, Wagholi, Pune, Maharashtra, India*
[***]*Bheemanna Khandre Institute of Technology, Bhalki, Karnataka, India*

(Corresponding author: Vipin Vibhute)
(Received 16 September, 2016 Accepted 19 October, 2016)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Learning from Demonstration (LfD) is an integral part in robot development. LfD is a technique used to teach robots various tasks. There are different types of LfDs and the prime object is making robots learn by mimicking or by observation. Reinforcement learning has gained importance over a period of robot development. Reinforcement learning is learning problems from situations. This paper introduces the types and methods reinforcement learning used in robots.

Keywords: Reinforcement Learning, Learning from Demonstration, Programming by Demonstration, machine learning, learning agents

I. INTRODUCTION

Robots are necessity of our future. Robots find place in all aspects of industry, in real life situations and in problem solving. But the greatest challenge faced in robot development is teach the robot real life and real time problems and embed intelligence in them to solve these problems.

This challenge to teach robots real life and real time problems gave birth to a wide new branch in robot development. The efforts and researches of scientists and technocrats opened the new technique - Learning from Demonstration (LfD). Evolution of LfD over the past few decades saw various methods of learning. With the support of mathematicians, Reinforcement Learning (RL) methods for robots have been in focus.

RL is a machine learning paradigm. Reinforcement learning is analogous to human learning, where in learning is through interactions with environment and outcome is reward based. The robot is not programmed to take any actions, but instead must learn by itself to find such actions which yield the most rewards.

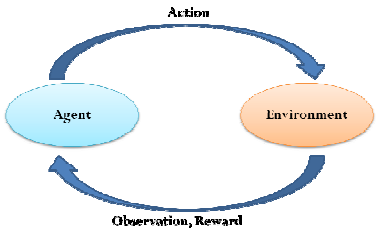

Fig. 1. Simple Reinforcement Learning.

RL principles: Program should study how to control an agent by self-trying out actions and use the studied feedback (rewards) to strengthen (reinforce) the agent's behavior. The agent interfaces with the environment by using its sensors, transducers and effectors. The reward is the one which determines which agent's behavior is most desired and acceptable.

Reinforcement learning is an adaptive interface between an active decision-making agent and its environment, within which the agent tries to achieve a desirable action despite of uncertainty in its environment. Reinforcement Learning has four main subelements: a policy, a reward function, a value function and a model of the environment (optional).

A *policy* governs the learning agent's behavior at a given time. It maps perceived states of the environment to actions to be taken when in those states. Mapping can be done by simple lookup table or in complex systems extensive computational search process.

A *reward function* is the achievement in reinforcement learning. It maps each perceived state of the environment to a single number, a reward, indicating the expected output of that state. A reinforcement learning agents' only objective is to maximize the total rewards it gains. The reward function defines meritorious and non-meritorious events for the agent.

A *value function* specifies meritorious states in the long run. The value of a state is accumulation of rewards that an agent can expect in the future, starting from that state. Whereas rewards are the immediate, intrinsic desired output of environmental states. The values are the long-term desired output of states after taking into account the states that are likely to proceed, and the rewards available in those states.

A *model* of the environment emulates the behavior of the environment.

Robot Learning from Demonstration initiated from the 1980s. LfD is inspired from the concepts of psychology and biology activities. Few LfDs follow a computational neuroscience approach and use neural modeling. Others use more cognitive science approach and conceptual model of imitation learning from animals. In Learning from Demonstration approach, preferences and parameters are automatically tuned to reproduce behavior that was demonstrated by an expert.

Reinforcement learning is the most adapted and now-a-days commonly used type of learning in robotics. Reinforcement learning is learning from trial and error.

II. LEARNING FROM DEMONSTRATION

Robots learn to perform tasks from demonstration by three different methods: mimicking, emulation and imitation.

Mimicking is copy& paste of demonstrators' bodily movements. Mimicking must not understand the purpose of the action instead just follow the demonstrator. So, through mimicking alone new tasks cannot be accomplished in uncertain environments.

In *emulation* learning, the robot replicates and duplicates the expected results of the demonstrators' action.

Imitation learning as a goal oriented mimicking. In imitation learning, an observer is expected to recognize demonstrator's movements and its output and use the expectations that to accomplish the task [2].

Fig. 2. Imitation Learning from Demonstratio.n
(Photo courtesy of QRIO by Sony)

Types of Learning from Demonstration

A robot can learn tasks in two ways – one by *imitation* and another by *sense-plan-act or teleoperation.* Imitation is by copying actions of the user and other is to sense the users' intention and then act accordingly.

Imitation Learning from Demonstration

Imitation learning is also called as *Learning by Observation*. A formal definition of imitation quoted by Mitchell (1987): *"... imitation occurs when something C (the copy) is produced by an organism and/or machine, where: C is similar to something else M (the model); registration of M is necessary for the production of C; and C is designed to be similar to M."*[1].

Imitation learning is used to acquire generic knowledge and use this knowledge to guide the robot. Imitation learning is adaptive and reuses the acquired knowledge in different contexts. Imitation learning teaches robot methods to sequence motion primitives in order to achieve complex tasks. Motion reproduction based on imitation learning.

In the robotics domain, imitation usually takes the form of *passive* and *active* observations of the teacher's activities. In passive imitation a robot is allowed to detect if the movements of the teacher match with any of its existing skills by testing the predictions of the models associated with those skills against the teacher's trajectory. If the teacher performs a movement that is not in the robot's repertoire, the passive imitation approach is employed to acquire the newly demonstrated behavior [1]. In active learning from demonstration the robot requests specific demonstrations from the teacher. Here the learner or robot is has to choose the training data or efficiently select a good training set. Active learning ensures that each demonstration is maximal useful and not redundant.

Teleoperation Learning from Demonstration

Teleoperation LfD is type of *Learning by Experience*. A teacher makes a robot perform and experience the tasks to be learned through its own sensors. The advantage of teleoperation is that the robot is not involved in interpreting and relating the actions of different teachers to its own. The acquired information from sensors is due to robots' own structure and sensory-motor skills. As this information is developed by itself, the problems of mismatched models are avoided. The teleoperation is accomplished using joysticks or other remote control devices, like haptic devices.

Teleoperating demonstration is performed by guiding the robot manually with a remote control or bya demonstrator performing the task without any intervention with the robot. While performing demonstration the motions are captured and recorded by means of a motion capturing system. The trajectory of a motionis directly recorded by motion capturing device or sensor and then it is replayed to the robot control to obtain the demonstrated motion. Teleoperation requires an interface for the direct control of all degrees of freedom. Teleoperation is the most direct method for teaching a robot. However, teleoperation's requisite is that operation of the robot should be manageable, and hence not all systems are suitable for this technique.

Teleoperation is advantageous as it allows the training of robots from a distance. So it is not necessary for the demonstrator to be near the robot. Hence it finds major role in teaching navigation and locomotion patterns.

III. REINFORCREMENT LEARNING (RL)

Features of RL: learn from data, learn a predictive model, learn to predict reward and learn a behavior that maximizes reward

Learning types in RL:

Passive: agent has a fixed policy (only learns how "good" each state or action is). Agent executes a fixed policy and evaluates it. The agent's action is based on a fixed policy and it tries to learn how good the policy is by observing the world. It is like evaluation of policy in policy iteration.

Active: An agent has to learn what to do and updates policy as it learns. The agent attempts to find an optima policy by exploring different actions in the world. It is like solving the underlying Markov Decision Processes (MDP).

Model-based: learn transition and reward model. It is used to getoptimal policy. It learns the MDP model (T and R), or an approximation of it.

Model free: Derive optimal policy without learning the model. It derives the optimal policy without explicitly learning the model.

Different types of reinforcement learning agents:

*Utility learning:*is learning utility function. The agent learns a utility function of states and uses this function to select actions that maximize utility of outcome. It selects actions such that utility is the maximum. The agent requires a model of state transitions.

Process of utility agent; for each possible action: the agent predicts the new state that it would take you to looks up the value of that state and then chooses the best.

Advantage that is uses "deeper" knowledge about domain. Disadvantage is that the agentmust learn model of environment. The agent alsoneeds to know where actions outcome in order to evaluate actions taken andmake decisions.

Q-learning: is learning action-value function. The agent learns expected utility for taking action in given state. It directly learns the utility of each action in a given state.

Advantage is that no model is required. Disadvantage is that agent has shallow knowledge, cannot look ahead and can restrict ability to learn.

Reflex agent: learning a policy that maps directly from states to actions. The agent just looks up your current state, sees what the best action is and does it.

Advantages are no model of the state transitions needed and no idea of utility needed.

IV. REINFORCEMENT LEARNING IN ROBOTS

Reinforcement learning is a general framework to convert traditional robotics to autonomy, versatility and adaptability [3]. In robotics, the ultimate goal of reinforcement learning is to enable and make the robots with the ability to learn, improvise, adapt and reproduce tasks. The ability is with dynamically changing constraints based on exploration and autonomous learning. An achievable way is by creating robots that can learn new skills by themselves analogous to humans[4].

Reinforcement learning is learning from trial and error. RL determines the ideal behavior of a robot by allowing machines and software agents to automatically interact, in order to maximize its performance. A feedback system is required for the agent to learn its behavior known as reward. This reward is the reinforcement signal. The learner is not instructed to which actions is to be taken, but instead must find out which actions yield the most reward by trial and error. This behavior can be one-time learning, or can timely keep on adapting.

Reinforcement learning (RL) is harder than supervised learning and it is the unique way to achieve very good desirable performance in many complex domains.

Reinforcement learning enables a robot to automatically find an optimal behavior through trial and error interactions with its environment. Reinforcement learning endows a robotics a framework and set of tools for the design of sophisticated and complicated behaviors.

Robot learns to control an agent by trial and error of actions and using the obtained rewards to reinforce the agent's behavior. The agent interacts with the environment by using its sensors and information extraction system.

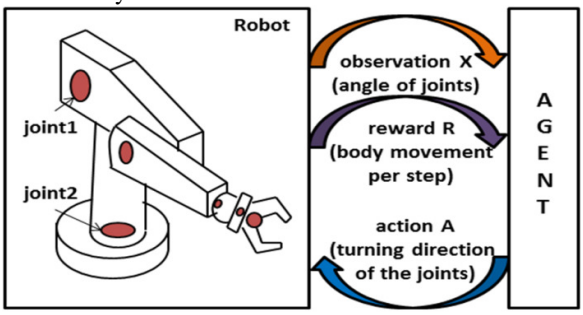

Fig. 3. Reinforcement learning in robots.

The reward function determines which agent's behavior is desirable. The agent may receive reward or punishment. The prime output of reinforcement learning is to discover an optimal policy that correlates states to actions. The optimal policy is used to maximize the expected return which relates to the cumulative expected reward. To obtain a large number of rewards, the reinforcement learning agent must select actions that was tried in the past and found to be optimal in producing reward.

All reinforcement learning agents must have pre-defined goals. It can detect aspects of their environments and choose optimal actions to take cognizance the environments. In practical applications, obtaining samples for reinforcement learning is often expensive.

Compared to reinforcement programming, dynamic programming is not applicable when:

- The Markov decision problem is not known.
- There are too many states to take decision.
- There are continuous changing states or actions.

Hence, reinforcement learning algorithms is used instead of dynamic programming algorithms.

The main advantages of reinforcement learning are:

1. A prior model of the environment is not needed. This is advantageousforin most complex tasks, a model of the environment isnot known or its description is too complex.
2. There is no necessity to know before hand to what actions for each situation must be presented to the learner.
3. The learning process is direct online by interacting with the environment.
4. It is capable of learning from zero start.

V. REINFORCED LEARNING FOR HUMANOID ROBOTICS

Humanoid Robots are very complex systems. They are multi-dimensional movement systems and require additional efforts and computation as compared to the conventional control problems. This learning method has to use data efficiently, able to upgrade for high dimensional continuous state and action plans. As well should be cheap in computation so as to work online. The method should also be easily analyzed through function approximation techniques[3].

Jan Peters et al stated in their research that greedy policy-improvement algorithms are likely to fail to upgrade for high dimensional movement systems. Because the large changes in the policy during learning creates difficulties in stable algorithms. They derived an algorithm, which approximates the natural gradient and secondly, the same natural gradient is used to improve the policy [3].

VI. REAL-TIME INTERACTIVE REINFORCEMENT LEARNING FOR ROBOTS

Interactive Reinforcement Learning is an approach for training robots which assist human by natural interaction. The reward signal is determined same as that determined conventionally in formal reinforcement learning. But in addition to this conventionally determined reward, rewards determined by a real-time interaction with a human teacher or coach are also considered. In an interactive reinforcement learning session, the human has option to change the reward signal not pertaining at certain goal states, but continuously throughout the interaction. However, interactive reinforcement learning also permits the human to remain passive when deferring to an independent environmental based reward signal. Interactive reinforcement learning emphasizes the *interactive* elements in teaching [5].

Andrea et al were able to identify a host of new considerations for reinforcement learning. Their experiments were directly from the introduction of a human-based real-time reward signal. Their results can be extended beyond robotic to other kinds of feedback-based learning systems [5].

VII. HYBRID REINFORCEMNT LEARNING

In hybrid reinforcement learning approach, a nonlinear function approximator is used. It is trained in batch mode on a dataset recorded while the policies which are externally supplied make management decisions within the system.

Neural networks are used because of their prior successes in reinforcement learning applications. They are robust in high dimensional spaces, so they may be used with other types of function approximators.

They use of an external policy so as to avoid poor performance that could be expected during online learning.

For a good initial policy it is necessary to use either a good external policy, or initialize the reinforcement learning value function. But, the second option is quite difficult, because it requires extensive domain knowledge of the particular system, along with intensive knowledge of the function approximator methodology [6].

Gerald Tesauro *et al,* conclude that hybrid reinforcement learning approach takes advantage of ability to learn in a knowledge free manner. This requires neither an explicit system model nor an explicit traffic model. It also requires little or no domain knowledge or its value function representation [6].

VIII. REWARD-WEIGHTED REGRESSION FOR REINFORCEMENT LEARNING

In reward-weighted reinforcement learning framework, the problem of learning with immediate rewards is reduced to a reward-weighted regression problem. It also incorporates an adaptive and integrated reward transformation for faster convergence. The outcome is efficient algorithm, smooth learning without dangerous jumps in solution space. Suitable for applications of complex robots with high degree-of-freedom.

Reward-weighted regression is efficient to deal with high-dimensional robot systems and huge amounts of data, with a low sample complexity. This regression guarantees convergence, and best suited for smooth and continuous online improvement [7].

Jan Peters at el developed a framework for learning operational space control. These types of controller had very less practical implementations due to issues with system identification in actual complex robots. The concept of reinforcement learning by reward-weighted regression is to find optimal solutions in reducing redundancy in operational space control [7].

IX. ROBOT REINFORCEMENT LEARNING USING CROWDS REWARDS

In this approach robot learning is done by combining basic skill transfer through physical demonstrations. Reinforcement learning forms an integral part with web-based crowd rewards. This approach teaches a robot the very basic motion skills through kinesthetic teaching. Crowd rewards teaches to manipulate several objects through reinforcement learning through the feedback from multiple non-expert crowd workers. By crowd rewards approach it is feasible to attain robot learning reducing the efforts of a dedicated teacher.

Crowds are online human feedback, both in real time and post hoc and used to improve the performance of learning systems. In the individual training of robots many problem spaces remain unexplored, which result in poor performance. This happens because appropriate policies are discovered after a large number of trial and error. In crowd based systems on demand human intelligence is exploited, to provide immediate feedback to reinforcement learning systems. The feedback is purely based on the experience and intuition of the human observer.

The online crowds can provide real time assistance to learning agents as and when the need arises. It is learned that crowd workers respond accurately and very fast enough to provide just in time feedback. It also shows that crowds can improve their accuracy in post hoc review settings and can use them in future conditions [8].

Gabriel V. de la Cruz Jr. et al conclude that crowd workers are able to accurately find mistakes in real time with a very less mean latency. The workers must also suggest an action [8].

X. CONCLUSION

Learning from demonstration (LfD) and reinforcement learning (RL) provide two approaches for an agent to learn how to accomplish a task. Using the above reinforcement learning techniques and the algorithms allow humans to more naturally communicate with interactive machine learning algorithms which will, ideally, make it easier for non-programmers to use these algorithms.

The future will address the evaluation of the learning framework in more complex tasks, in particular in conjunction with priming a control policy from imitation learning and subsequent self-improvement.

REFERENCES

[1] Monica Nicolette Nicolescu, “a framework for learning from demonstration, generalization and practice in human-robot domains,” in *Dissertation of Doctor of Philosophy*, May 2003.

[2] Baris Akgun, Kaushik Subramanian, Andrea L. Thomaz, “Novel Interaction Strategies for Learning from Teleoperation,”*Georgia Institute of Technology College of Computing*, 801 Atlantic Dr., Atlanta, GA.

[3] Jan Peters, Sethu Vijayakumar, Stefan Schaal, “Reinforcement Learning for Humanoid Robotics”Humanoids2003, *Third IEEE-RAS International Conference on Humanoid Robots*, Karlsruhe, Germany, Sept.29-30.

[4] Petar Kormushev, Sylvain Calinon and Darwin G. Caldwell, “Reinforcement Learning in Robotics: Applications andReal-World Challenges”, in *Robotics 2013, 2, 122-148*; doi:10.3390/robotics2030122, www.mdpi.com/journal/robotics.

[5] Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal, “Real-Time Interactive Reinforcement Learning for Robots,” in *American Association for Artificial Intelligence*, (www.aaai.org), 2005.

[6] Gerald Tesauro, Nicholas K. Jongt, Rajarshi Das and Mohamed N. Bennanit, “A Hybrid Reinforcement Learning Approach to Autonomic Resourch Allocation,” in *IEEE Transaction 1-4244-0175-5/06*, 2006.

[7] Jan Peters, Stefan Schaal, “Using Reward-weighted Regressionfor Reinforcement Learning of Task Space Control,” in *Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007) IEEE*.

[8] Gabriel V. de la Cruz Jr., Bei Peng, Walter S. Lasecki , Matthew E. Taylor“Generating Real-Time Crowd Advice to Improve,” in *American Association for Artificial Intelligence*, (www.aaai.org), 2015.