Q Learning Based Trajectory Generation for Robotic Grinding and Polishing

Gbenga Abiodun Odesanmi[a], Imran Iqbal[b], Bai Jie[c], Zhang Cong[d], Jianxiang Wang[e], and Li Michael Liu[f]

Center of Excellence for Intelligent Mechanical Systems,
Department of Mechanics and Engineering Science, College of Engineering,
Office Add: 902 Wang Ke Zhen Building, Peking University, Beijing 100871, China,

[a]gbengaodesanmi@pku.edu.cn, [b]imraniqbalrajput@pku.edu.cn, [c]baijierobot@163.com,
[d]zhangc@sci-robot.com, [e]baijierobot@163.com, [f]jxwang@pku.edu.cn liuli@coe.pku.edu.cn

Keywords: Robot Manipulation, Machining Operation, Trajectory Generation, Q-Learning;

Abstract. In this paper, we apply Q learning algorithm as learning method for grinding and polishing operation. Our goal is to make the robot manipulator to learn the optimal grinding and polishing path trajectory for any work piece. The performance from our simulation and experiment shows that a robotic manipulator with Q-learning algorithm proposed in this paper was able to learn the optimal path trajectory for the whole work piece surface and choose the best decision in each situation of its dynamic environment.

Introduction

Recent application of reinforcement learning methods have proved to be successful in various robotic control tasks like locomotion [1,2], manipulation [3,4,5,6,7] and autonomous vehicle control [8]. Many organizations are working intensively on how to improve the possibility of building intelligent mechanical systems like robot that can sense, plan their own motions, and execution as contributed to the realization of the intelligent systems.

Grinding and polishing operations are applied in variety of fields ranging from industrial applications like machining operations on metals or different materials to medical applications like dentistry and orthopedics [9].

With the aim of achieving good surface finishing with rapid development of industrial techniques, the digital machining operation turn to be more complex and requires increasingly high precision and accuracy, which make the traditional manual machining inefficient because grinding and polishing operation required high skilled labor and [10] the dust is extremely harmful for human.

Application of robotic system to machining operation is achieving a high degree of accuracy and material removal rate in the parts with complex geometries, and presents a significant advance in the state of the arts in surface finishing which is responsible for 10 to 30 percent of total manufacturing cost [11,12]. To achieve good surface finishing there is need for efficient and accurate control system for the robot to yield good performance in machining.

Small deviations of either the tool attached to the robot end effector or the work piece would result in process failure, as the non-flexible programming of the motion is not able to deal with small variations of the environment.

This general problem gives rise to the question how robots can be enabled to deal with such variations of the environment and autonomously adapt to them to plan their motion in a flexible way [13].

In the light of this, many approaches have been proposed, investigated and employed for robotic manipulator and control [13], in robot machining, it is essential to control the tangential velocity of the tool along the work piece and the force normal to the work surface.

It is very difficult to achieve an acceptable system performance because of high level of unpredictable interaction between the robot and the environment. Application of reinforcement learning offers robotics a framework and set of tools for the design of hard to engineering behavior while enabling robot to autonomously discover an optimal behavior through trial and error interaction with its environment [14]. The objective of this studies is to use reinforcement learning algorithm to learn an optimal path for robotic grinding and polishing. To our knowledge, reinforcement learning has not been applied to such task.

The contribution of this paper is to establish a simple motion control method based on reinforcement Learning algorithm for grinding and polishing operation and to learn the optimal motion path base on the shape of the work piece.

Related work

Reinforcement Learning prove to be a convenient method to learn complex controls task without the prior knowledge of the dynamics of the environment [15] and has been demonstrated in many fields, such as game theory, control engineering, statistics, or even robotics when toy models or very low-realistic robot simulators are used [16,17,18,19].

However, [20,21,22,23,24] proposed considerable work for continuous control of robotic of high dimensional system based on general purpose neural network with reinforcement learning and [25,26] work utilizes deep reinforcement learning to solve a wide variety of continuous motor control problems like motion planning for industrial robots directly from sensory input.

All these approach shows the conceptual usability of reinforcement learning methods which rely on heavy computational effort both in terms of the number of used cores and the required computation time. Basically, the goal of the prior work is fundamentally different from the scope of this work. Our aim is to make the robotic arm learn the shortest way of accomplishing grinding and polishing task and also taught how to determine the optimal grinding and polishing path based on the shape of the work piece. In this case we will use classic Q learning as our learning techniques.

Background

In this section, we define the problem of grinding and polishing as a Markov decision process whereby the agent learns a policy (π) which maps a sequence of states (x_t) into a sequence of actions (u_t) in the form of torque applied to the robot's joints. The goal in reinforcement learning is to control an agent attempting to maximize a reward function which satisfy the user provided definition of what the robot should accomplish. This is done by direct interacts with the environment by taking an actions to maximize the cumulative reward. At each observation, the agent receives a reward (r_t) which can either be positive or negative at each state visited based on the action selected [27].

Model Formulation. Q-learning [27] is a model-free learning method that can also be viewed as asynchronous dynamic programming (DP) method which provides agents with the capability to learn an optimal policy in Markov Decision process by experiencing the consequences of actions, without requiring them to build maps of the domains. Furthermore, a policy is a deliberate system

of principles to guide decisions or more specifically, a decision function that specifies what the agent will do for each possible value that it can sense [28]. The agent need to take into account not only the immediate reward but also possible future rewards.
The discounted future returns can be given as by:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{n-t} r_n \quad (1)$$

Where $\gamma \in (0,1]$ is the discount factor and it determines how strong the agent takes future rewards into account. When $\gamma = 0$ represent a short-sighted strategy as higher-order terms for rewards in the future become negligible. If the environment is deterministic, γ can be set to 1 as the same actions always result in the same rewards. A good strategy for an agent trying to maximize its discounted future reward can be learned using Q-learning method [13]. Representing our Q-learning as a function of state and action we have:

$$Q(x_t, u_t) = \max(R_{t+1}) \quad (2)$$

As stated above, the aim of reinforcement learning agent is to find optimal policy, the value of the action (u) performed in state(x) at time step t can be represented as:

$$\pi(x) = argmax_u\big(Q(x,u)\big) \quad (3)$$

After many iterations the Q-value can be estimated as Bellman equation [30], and can be written as follow:

$$Q(x,u) = r + \gamma . max_{u'}\big(Q(x',u')\big) \quad (4)$$

Among the off-policy methods, Q-learning offers significant data efficiency compare to on-line policy which is crucial for robotic application [29]. Over many training, the update of the Q-value for each state and action pair can be done to generate a q-table using the following formulation:

$$Q(x,u) = Q(x,u) + \propto \Big(r + \gamma . max_{u'}\big(Q(x',u')\big) - Q(x,u)\Big) \quad (5)$$

where α is the learning rate which determines the difference between the previous Q-value and the newly proposed Q-value is taken into account. Note that for α =1, Eq.5 simplifies to the Bellman Eq.4. It has been shown, that this iterative approach of estimating the Q-function converges and, given enough iterations, represents the true Q-value [30].

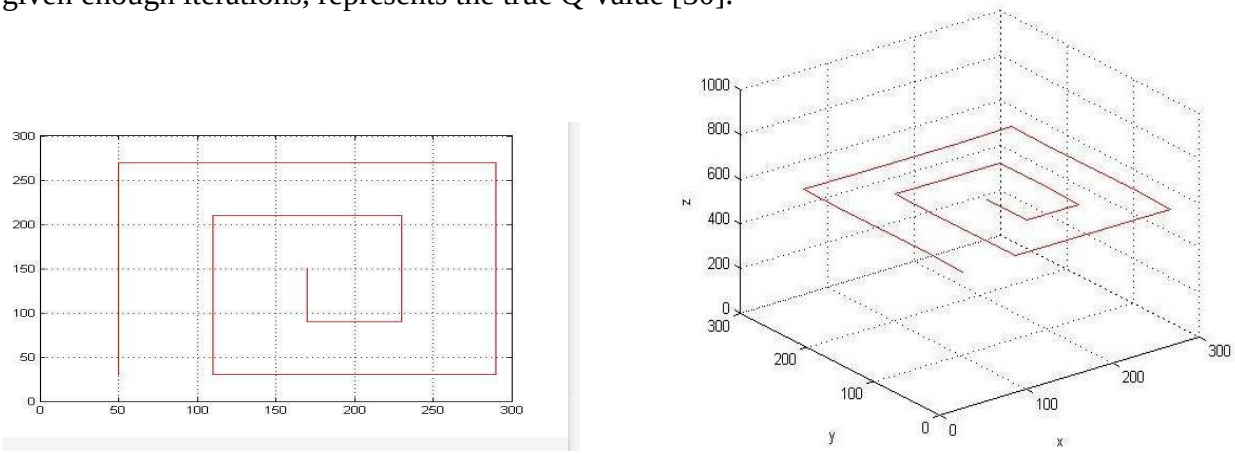

Fig. 1: 2D (left) and 3D (right) grinding and polishing motion trajectory of a simple 300 by 300mm plate surface.

Experiment and Results

In this work, the learning procedure of the model was implemented in virtual robot experimentation platform(V-REP) [31] and was interacted remotely with V-REP remote API client library which was used as alternative to ROS interface. This serve as our environment, it consists of the robot arm, grinding tools and the work piece. The learning agent is an entity which control the robot arm, it implements the learning algorithm, choosing the actions, obtain the new state and receive the rewards. Our goal is to make the robot arm learn a simple and optimal motion for grinding and polishing operation. The grinding surface was segmented and the reward matrix was set and the action is to move the arm along the work piece surface (i.e. up, down, left or right).

However, in order to ensure the conformity between the robot and the work piece, we set the tips of the end effector to the starting point of the work piece. From the initial state, the agent starts to explore its environment by performing actions, triggering state transitions in the environment and gathering corresponding rewards. As the learning proceeds, the agent explores its environment, it stores experiences as state, action, reward and iteratively estimates the Q-values for each state and action pairs encountered during the training. The agent starts with no experience about the environment and it will perform random actions every time it find itself in an unknown state. As the agent gathers experience over series of iteration, the agent starts to update the Q-value and learns how to continuously generate the optimal grinding and polishing path motion.
Fig.1 shows the learned 2D and 3D motion trajectory generated for the work piece surface by the while Fig.2 shows the experimental results, the x-axis shows the number of episode and the y-axis represent the rewards received by the agent. The result demonstrated that increase in the number of training increases the rate of convergent. The learning performance proves that the introduced approach is feasible for robotic grinding and polishing.

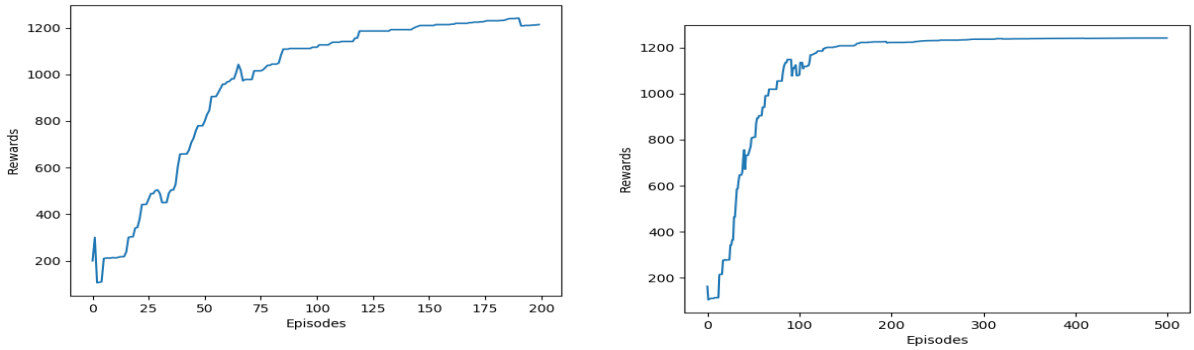

Fig.2: Learning curve for the grinding and polishing motion trajectory with 200 and 500 episodes.

Q learning Algorithm

- Initialize the value of Q(x,u) arbitrarily
- At each time the agent chooses an action and observe its reward.
- The agent updates its Q-value base on Q-Learning update rule
- Set the next state as the current state
- End

Discussion and Future Works

In this paper, we presented a Q-learning approach that can be used to learn a complex robotic manipulation skills and we applied this method to machining operation in which the agent was

able to learn the complete grinding and polishing motion for the whole surface of the work piece within a few minutes of learning. Our simulation results confirmed that a robotic manipulator with Q-learning algorithm proposed in this paper was able to learn and choose the best decision in each situation of its dynamic environment. Although, this is just the first stage of our work, in the future work, we will investigate other reinforcement learning method with function approximation method like neural network.

Acknowledgements

This work was supported by the "Chinese Natural Science Research Project" under grant "2017YFC0110700".

References

[1] N. Kohl and P. Stone, Policy gradient reinforcement learning for fast quadrupedal locomotion. International Conference on Robotics and Automation (IROS), 2004.
[2] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, and G. Cheng, Learning CPG-based biped locomotion with a policy gradient method: Application to a humanoid robot, International Journal of Robotic Research, vol. 27, no. 2, pp. 213–228, 2008.
[3] Gu, S., Holly, E., Lillicrap, T. P., & Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. international conference on robotics and automation, 3389-3396, 2017
[4] J. Peters, K. Mulling, and Y. Alt ¨ un, "Relative entropy policy search, AAAI Conference on Artificial Intelligence, 2010.
[5] E. Theodorou, J. Buchli, and S. Schaal, Reinforcement learning of motor skills in high dimensions, International Conference on Robotics and Automation (ICRA), 2010.
[6] J. Peters and S. Schaal, Reinforcement learning of motor skills with policy gradients, Neural Networks, vol. 21, no. 4, pp. 682–697, 2008.
[7] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, Learning force control policies for compliant manipulation, International Conference on Intelligent Robots and Systems (IROS), 2011.
[8] P. Abbeel, A. Coates, M. Quigley, and A. Ng, An application of reinforcement learning to aerobatic helicopter flight, Advances in Neural Information Processing Systems (NIPS), 2006.
[9] A. Balijepalli and T. Kesavadas. A Haptic Based Virtual Grinding Tool. Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperation Systems (HAPTICS'03).0-7695-1890
[10] Yaonan Li, Heping Chen and Ning Xi. Automatic Programming for Robotic Grinding Using Real Time 3D Measurement. The 7th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems.2017
[11] N. Ramachandran, N. Pande, and N. Ramakrishnan, The Role of Deburring in Manufacturing: A State-of-the-Art Survey, Jouml of Materials Processing Technology, vol. 44, 1994.
[12] R. Komanduri, M.E. Merchant, and M.C. Shaw, Symposium on US Contributions to Machining and Grinding Research in the 20th Century,Applied Mechanics Reviews, vol. 46, no. 3, 1993.
[13] Richard Meyes, Hasan Tercan, Simon Roggendorf, Thomas Thiele, Christian Büscher, Markus Obdenbusch, Christian Brecher, Sabina Jeschke, Tobias Meisen. Motion Planning for

Industrial Robots using Reinforcement Learning. The 50th CIRP Conference on Manufacturing Systems doi: 10.1016/j.procir.2017.03.095 Procedia CIRP 63 (2017) 107 – 112
[14] C. Yang, J. Luo, Y. Pan, Z. Liu, C. Su, Personalized variable gain control with tremor attenuation for robot teleoperation, IEEE Trans. Syst. Man Cybern. Syst. (2018). in press. doi: 10.1109/TSMC.2017.2694020.
[15] Arun Kumar, Navneet Paul, S N Omkar. Bipedal Walking Robot using Deep Deterministic Policy Gradient, arXiv:1807.05924v2 [cs.RO] 17 Jul 2018
[16] Wiering, M. & Van Otterlo, M. "Reinforcement learning: State-of-the-art', Adaptation, Learning, and Optimization". 2012.
[17] Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey, Journal of Artificial Intelligence Research 4, 237–285. 1996
[18] Kober, J., Bagnell, J. A. & Peters, J. Reinforcement learning in robotics: A survey, The International Journal of Robotics Research p. 0278364913495721. 2013
[19] Sutton, R. S. & Barto, A. G. Reinforcement learning: An introduction, Vol. 1, MIT press Cambridge. 1998
[20] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, Neural ˙ networks for control systems: A survey, Automatica, vol. 28, no. 6, pp. 1083–1112, Nov. 1992.
[21] M. Riedmiller, Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method, European Conference on Machine Learning. Springer, 2005, pp. 317–328.
[22] R. Hafner and M. Riedmiller, Neural reinforcement learning controllers for a real robot application, International Conference on Robotics and Automation (ICRA), 2007.
[23] M. Riedmiller, S. Lange, and A. Voigtlaender, Autonomous reinforcement learning on raw visual input data in a real world application, International Joint Conference on Neural Networks, 2012.
[24] J. Koutn´ık, G. Cuccu, J. Schmidhuber, and F. Gomez, Evolving largescale neural networks for vision-based reinforcement learning, Conference on Genetic and Evolutionary Computation, ser. GECCO '13, 2013.
[25] Sergey Levine, Chelsea Finn, Trevor Darrell and Pieter Abbeel, End-to-End Training of Deep Visuomotor Policies, Journal of Machine Learning Research 17(39): 1- 40, 2016
[26] Volodymyr Mnih, Adrià Puigdomènech, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, arXiv preprint arXiv:1602.01783, 2016
[27] Watkins, C. Learning from Delayed Rewards, PhD thesis, University of Cambridge, Cambridge, UK. 1989.
[28] Poole D.L. and Mackworth A.K. Artificial Intelligence: foundations of computational agents, Cambridge University Press. 2010
[29] Bellman R. (1952) On the theory of dynamic programming. Proceedings of the national Academy of Sciences,38(8):716-719.
[30] Melo F.S., Convergence of Q-learning: A simple proof, http://users.isr.ist.utl.pt/~mtjspaan/readingGroup/ProofQlearning.pdf (last checked at 28.11.2016).
[31] E. Rohmer, S. P. Singh, and M. Freese, V-rep: A versatile and scalable robot simulation framework, Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, 2013, pp. 1321–13