Journal of Korea Robotics Society (2020) 15(4):398-403
https://doi.org/10.7746/jkros.2020.15.4.398 ISSN: 1975-6291 / eISSN: 2287-3961

공 던지기 로봇의 정책 예측 심층 강화학습

Deep Reinforcement Learning of Ball Throwing Robot's Policy Prediction

강 영 균[1] · 이 철 수†
Yeong-Gyun Kang[1], Cheol-Soo Lee†

Abstract: Robot's throwing control is difficult to accurately calculate because of air resistance and rotational inertia, etc. This complexity can be solved by using machine learning. Reinforcement learning using reward function puts limit on adapting to new environment for robots. Therefore, this paper applied deep reinforcement learning using neural network without reward function. Throwing is evaluated as a success or failure. AI network learns by taking the target position and control policy as input and yielding the evaluation as output. Then, the task is carried out by predicting the success probability according to the target location and control policy and searching the policy with the highest probability. Repeating this task can result in performance improvements as data accumulates. And this model can even predict tasks that were not previously attempted which means it is an universally applicable learning model for any new environment. According to the data results from 520 experiments, this learning model guarantees 75% success rate.

Keywords: Robotic Arm, Deep Reinforcement Learning, Ball Throwing

1. 서 론

물체의 던지기는 안정성과 정확도가 보장되는 경우 효과적인 물체 이송방법 중 하나이다[1]. 물체의 던지기를 로봇이 수행할 때 포물선 운동을 가정한 운동학적인 해석을 통해 제어할 수 있지만 공기 저항, 물체의 회전 관성 등 정확한 계산이 어려운 변수들이 존재한다. 또한 로봇의 기하학적 정보들을 필요로 한다는 단점이 있다. 이러한 로봇의 던지기 수행에 있어 한계점을 극복하고자 기계학습을 도입한 제어 연구가 진행 중이다[2-4].

로봇의 작업 수행 환경과 역학적인 측정 및 계산이 어려울 경우 강화 학습을 적용해 작업을 수행할 수 있다. 강화학습은 작업 수행 능력을 점수화 하여 더 높은 점수를 도출할 수 있는 방향으로 로봇을 제어하는 학습방식이다. 작업 목적에 부합하는 제어에 대해 높은 점수를 부여하게 된다. 이러한 강화학습의 특성을 이용해 작업 환경에 대한 정보나 사물 간의 상호작용에 대한 정확한 측정, 계산없이 로봇은 주어진 목적에 부합하는 작업 수행을 할 수 있다[5,6].

강화학습을 이용한 공 던지기, 다르 던지기, 기는 로봇 등 다양한 주제에 대해 보상 값 평가 방식, 경사하강법 등을 적용한 연구가 이루어지고 있다[2,7,8]. 하지만 작업 환경에 대한 정보가 부족하거나 작업 수행 능력을 평가할 지표가 수학적으로 정립되지 않는다면 보상함수의 설계와 평가가 어렵다[3]. 또한 보상함수의 설계는 사용자 주관에 따라 설계가 상이 할 수 있다[9]. 이러한 한계점을 극복하고자 작업의 성공과 실패를 평가하는 방식이 도입되었다. 이 방식은 정보량은 적으나 실용적이고 복잡한 측정이 필요 없다[4].

본 논문에서 로봇이 학습할 작업 목표위치가 변하는 공던지기다. 보상함수의 설계 없이 골대의 센서를 통해 작업의 성공 또는 실패를 평가한다. 이 결과를 출력으로 하고 골대의 위치와 각 로봇 관절의 제어 정책을 입력으로 하는 인공지능망을 학습시켜 골대의 위치와 제어에 따른 작업 성공 확률을 학습하게 된다. 이후 목표위치에 공을 던질 때 가장 성공확률이 높은 제어 정책을 탐색해 수행한다. 이를 통해 보상함수의 설계 없이 성능의 향상시키며 작업을 학습할 수 있다.

로봇이 수행할 작업과 학습 방법에 대해 2장에서 기술한다.

Received : Sep. 2. 2020; Revised : Sep. 21. 2020; Accepted : Sep. 22. 2020
1. MS Student, Mechanical Engineering, Sogang University, Seoul, Korea (kyk6230@sogang.ac.kr)
† Professor, Corresponding author: Mechanical Engineering, Sogang University, Seoul, Korea (cscam@sogang.ac.kr)

3장에서는 로봇을 포함한 실험 환경과 기계학습 모델과 학습 알고리즘 대해 기술한다. 4장에서는 로봇의 공 던지기 강화 학습 실험의 과정과 결과에 대해 기술한다.

2. 이론적 배경

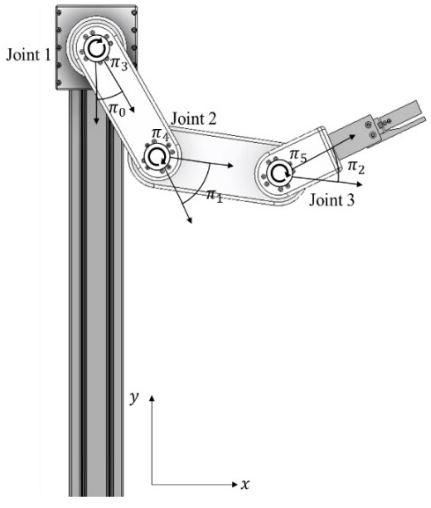
2.1 물체의 던지기

로봇의 제어에 있어 운동학적인 해석을 통한 제어가 가능하다. 물체를 비스듬하게 던져 올려 원하는 위치에 도달시키기 위해선 포물선 운동을 가정하고 물체를 던질 수 있다. 하지만 공기 저항, 물체의 회전 관성, 잡는 위치의 변화 등 물리적인 해석으로 로봇을 제어하여 원하는 위치에 도달시키기에는 여러 변수들이 있다[3,10]. 또한 로봇의 기하학적 정보가 필요하다. 이러한 운동학적인 해석의 한계점을 극복하고자 물체 던지기 작업에 기계 학습을 도입하였다.

로봇의 던지기 수행 강화학습에 있어 작업에 대한 평가가 필요하다. 선행 연구에서 사용한 보상함수 설계는 각 수행 마다 점수로 평가하며 이는 작업 환경에 대한 정보가 필요하다. 본 논문에서는 작업환경에 대한 정보를 최소한으로 학습에 사용하기 위해 성공과 실패 이분법적인 평가 방식을 도입하였다. 로봇의 입력 대비 제어 각도, 각속도, 골대의 위치, 골대의 센서를 이용한 작업 성공 판별만을 이용하여 학습을 진행한다.

2.2 동작의 변수화

본 논문에서는 [Fig. 1]과 같은 3자유도 로봇 팔을 이용해 실험을 진행하였다. 각 관절의 던질 때의 각도를 (π_0, π_1, π_2), 각속도를 (π_3, π_4, π_5)라 한다. 이 6개의 변수$(\pi_n \ \ (n=0,1,2\cdots 5))$가 제어 정책으로 학습된다. 이는 직관적이며 학습의 입력 변수로서 다루기 쉽다. 던지기를 실행할 시점의 각도, 각속도가 결정되면 [Fig. 2]의 속도 프로파일의 면적은 각도이기 때문에 관절의 구동 시점을 계산할 수 있다. 가속도는 일정하게 지정한다. 이 과정을 통해 입력한 각도, 각속도가 될 때 그리퍼를 작동시켜 물체를 던질 수 있다[11].



[Fig. 1] 3 DoF robot arm

2.3 학습 방법

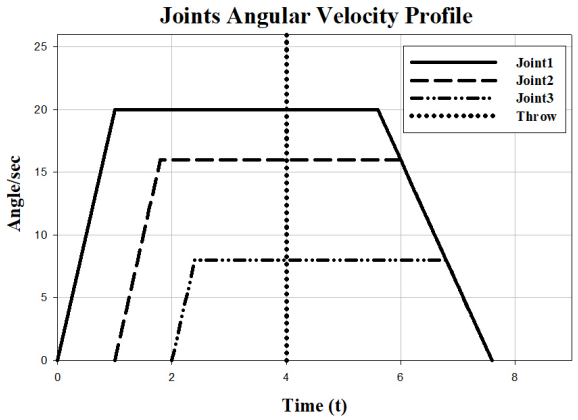
기존의 강화학습은 보상함수를 이용한 경사 하강법과 같은 최적화 기법들을 사용하였다. 이는 로봇 작동 환경에 대한 기하학적 정보 혹은 수행 작업에 대한 보상 함수를 필요로 한다[2]. 하지만 작업에 대한 정보가 불충분하거나 복합적인 파라미터들이 존재한다면 보상함수의 설계는 어려운 일이다[4].

2.3.1 심층 강화 학습

본 논문에서 제안하는 강화학습은 인공지능망을 이용한 성공확률 예측에 기반한다. 기존의 강화학습은 연속적인 함수의 형태로 작업 수행 능력을 평가했다. 인공지능망은 작업의 성공과 실패를 출력으로, 작업의 제어 정책과 골대의 위치를 입력으로 학습한다[12,13]. 작업 수행 결과는 성공과 실패, 이분법적으로 평가하며 1또는 0의 데이터로 저장된다. 이를 통해 인공지능망은 골대의 위치와 제어 정책을 입력하면 작업의 성공확률을 도출하게 된다.

2.3.2 정책 예측

골대의 위치가 정해지면 로봇이 수행 가능한 움직임 내에서 무작위의 제어 정책을 생성한다. 이때 생성된 제어 정책들은 인공지능망을 이용해 해당 목표위치에 도달할 수 있는 성공확률이 계산된다. 이 과정을 통해 도출된 성공확률 중 최고값을 갖는 제어 정책으로 로봇은 작업을 수행한다.



[Fig. 2] Joint angular velocity profile

3. 연구 방법

3.1 실험 환경 구성

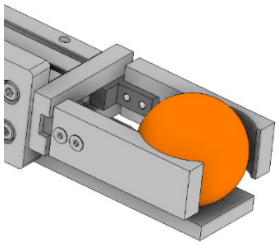
3.1.1 로봇 설계 및 특성

본 실험에 사용된 [Fig. 1]의 3자유도 로봇은 각 관절은 같은 평면상에서 움직이며 사람의 어께, 팔꿈치, 손목에 해당하는 역할을 수행한다. 실험이 안전하게 진행되고 인간의 관절 움직임이 가능한 내에서 각 모터는 [Table 1]과 같이 구동 제한을 두었다. 물체를 잡는 로봇의 그리퍼는 [Fig. 3]의 공압 그리퍼를 사용하였으며 공의 양 옆쪽을 잡고 놓을 수 있으며 아래를 받치고 있다.

[Fig. 4(a)]은 골대 로봇으로 2자유도 평형로봇을 사용했다. 로봇 바닥 중간 지점을 원점으로 지정 시 골대의 구동 범위는 [Table 2]와 같다. MPU6050 가속도센서가 골 망에 부착되어 있어 공이 골대에 들어가게 되면 센서와 충돌하게 되고 PC에 골 신호를 보내게 된다. [Fig. 4(b)]는 로봇 팔과 골대 로봇을 설치한 전체 실험 시스템이다. [Fig. 5]는 실제 실험 환경이다.

[Table 1] Driver Limitation

Joint	π_0, π_1, π_2		π_3, π_4, π_5
	Min Angle	Max Angle	Max RPM
1, 2, 3	0	60 degree	60 RPM



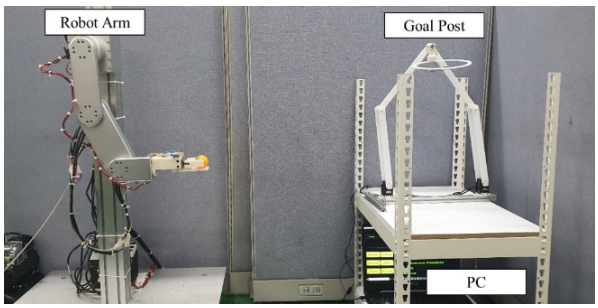
[Fig. 3] Gripper

(a) (b)

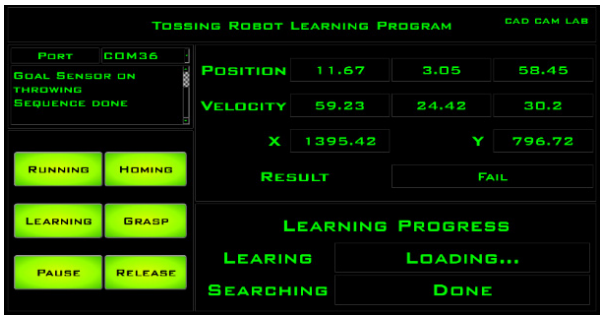
[Fig. 4] (a) Parallel goal post parallel robot, (b) Overall System

[Table 2] Goal Post Position Range

(mm)	Min	Max
X	-200	200
Y	470	310



[Fig. 5] Experimental environment



[Fig. 6] Software

3.1.2 실험 소프트웨어 시스템

[Fig. 6]는 본 실험에서 사용된 소프트웨어는 Python 기반으로 제작되었다. TensorFlow를 이용한 기계학습 모델을 사용하였으며 PyQt5를 이용한 UI를 제작하여 사용하였다.

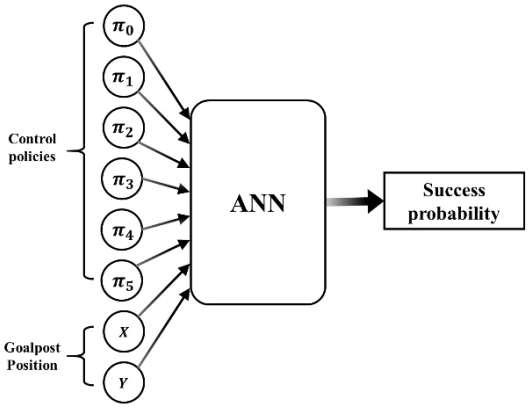
3.2 기계학습 모델

3.2.1 학습 모델 구조

학습에 사용된 인공지능망은 입력 층(8-D), 은닉 층, 출력 층(1-D)으로 구성된다[Fig. 7]. 입력 층의 제어 정책과 골대의 위치는 구동 제한 값을 이용해 정규화(Normalization)을 거친 후 학습된다. 은닉 층은 총 3개의 층으로 이뤄져 있으며 [Table 3]와 같은 구성을 갖는다. 이 보다 적은 parameter 수를 갖는 은닉층을 사용한다면 성공데이터에 대한 성공확률 값이 낮게 예측되었다. 현 물리적 현상을 대변하는 모델을 형성하기에는 parameter가 부족함을 나타낸다. 반대로 보다 많은 parameter 수를 갖게 되면 실패 데이터 주위에서도 높은 성공확률이 예측된다. 이는 같은 성능대비 더 많은 데이터를 요구한다는 결과이다. 각 은닉 층 사이에 Batch normalization을 통해 초기값에 대한 의존성을 낮추도록 했다. Dropout의 값은 0.2으로 과적합을 막기위함이다. 출력은 0~1사이의 값을 갖게 하기 위해

Sigmoid 활성 함수를 사용하였다. 성공과 실패와 같은 이분법적인 출력 값을 학습할 때 효과적인 Binary Cross-Entropy (BCE)를 Loss로 사용하였다.

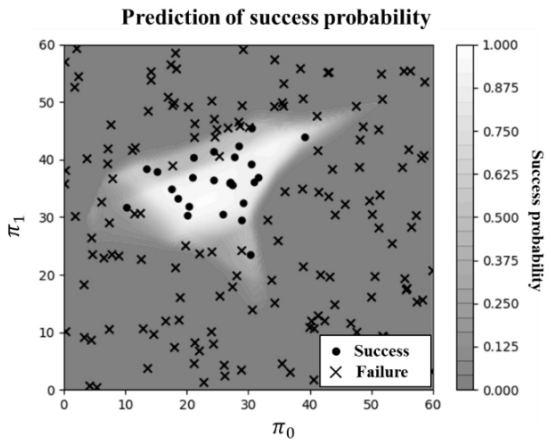
[Fig. 8]은 학습에 사용된 데이터 중 π_0, π_1에 따른 성공과 실패를 Scatter plot으로 표시하였다. 학습된 모델에서 π_0, π_1 이외의 값을 고정한 후 π_0, π_1의 변화 따라 예측한 성공확률을 등고선을 이용해 나타냈다. 성공한 데이터가 많은 조합일수록 성공확률이 높게 예측된 것을 확인할 수 있다. 이러한 예측 정책의 학습을 통해 골대의 위치에 따른 성공확률이 높은 제어 정책을 탐색하는 것이 가능하다. 작업에 대한 단순하고 직관적인 평가만을 통해서 로봇은 작업 수행 능력을 상승시키는 방향으로 학습될 수 있다.



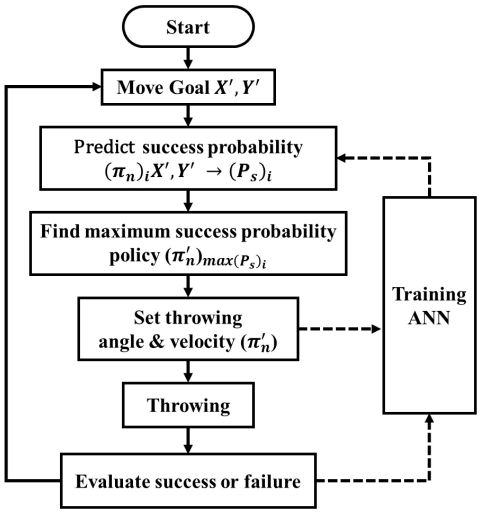
[Fig. 7] Neural network model

[Table 3] Neural network layer

Layer (type)	Output Shape	Number of Parameter
dense_1 (Dense)	64	576
batch_normalization_1	64	256
dense_2 (Dense)	64	4160
batch_normalization_2	64	256
dropout_1 (Dropout)	64	0
dense_3 (Dense)	32	2080
batch_normalization_3	32	128
dense_4 (Dense) Output	1	33



[Fig. 8] Prediction of success probability



[Fig. 9] Learning algorithm

3.2.2 학습 알고리즘

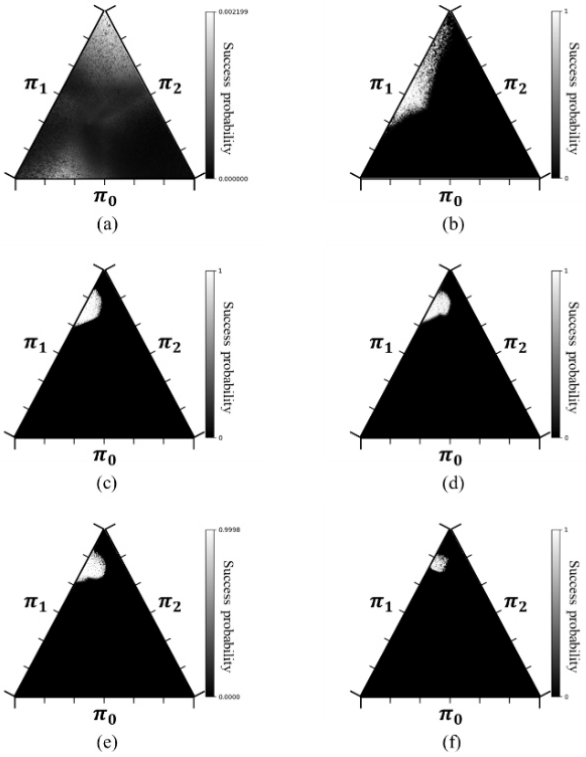
[Fig. 9]는 학습 알고리즘의 Flowchart이다. 작업 결과를 출력으로, 수행한 제어 정책과 골대의 위치(π_n', X', Y')를 입력으로 데이터를 저장한다. 공 던지기가 실행된 이후 성공이라면 1, 실패라면 0으로 출력 데이터를 저장한다. 이러한 학습 데이터를 통해 제어 정책과 골대의 위치에 따른 던지기의 성공확률이 학습된다.

처음 수행되는 던지기 이외의 모든 작업들은 다음의 탐색 과정을 거친다. 골대의 위치(X', Y')가 지정되고 총 1,000,000개의 무작위 제어 정책 쌍$(\pi_n)_i$ $(i = 1,2,3 \cdots 1{,}000{,}000)$을 생성한다. 이전 수행 데이터로 학습된 모델을 통해 골대 위치 (X', Y')에 대한 제어 정책 쌍 $(\pi_n)_i$의 성공 확률 $(P_s)_i$을 도출한다. 이때 가장 높은 성공 확률을 갖는 제어 정책 $(\pi'_n)_{\max(P_s)_i}$으로 던지기를 수행한다. 다음 던지기 수행 때에 이전에 학습된 모델을 이용해 성공 확률을 예측하여 제어 정책을 탐색한다.

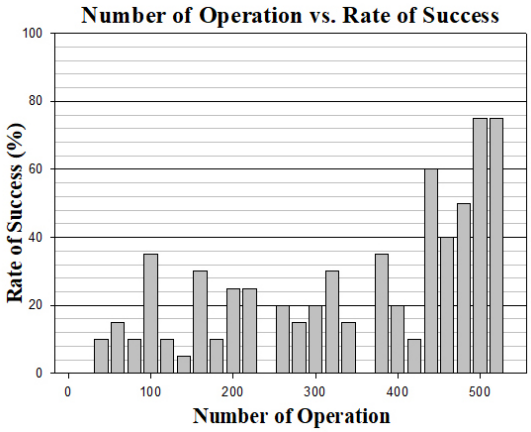
4. 실험 결과

[Fig. 10]에서는 설명을 위하여 목표 위치와 각속도 값을 고정하고 각도가 무작위인 제어 정책을 생성해 학습된 인공지능망을 통해 성공확률을 예측 이를 Ternary 그래프를 이용해 성공확률 예측을 나타냈다. 학습 초기 로봇은 높은 확률로 실패

하게 되고 이에 대한 성공확률을 '0'으로 학습하게 된다. [Fig. 10(a)]는 단 한번도 성공하지 못한 상태이다. 하지만 실패 데이터가 축적될수록 실행하지 않은 미지의 제어 정책에 대해서는 상대적으로 높은 확률이 예측된다. [Fig. 10(b)]는 50번 작업을 수행했으며 3번의 성공 데이터가 있다. 특정 정책에 대한 성공확률이 상대적으로 높게 예측되는 것을 확인할 수 있다. [Figs. 10(c)~(f)]에서는 더 많은 데이터가 축적됨에 따라 성공 확률이 높게 측정되는 정책의 범위가 좁아지며 기계학습 모델의 정책 예측 성능이 향상됨을 알 수 있다.



[Fig. 10] Prediction of success probability by number of data (a) 20data (no success data), (b) 50 data, (c) 100 data, (d) 200data, (e) 300data, (f) 500data (π_3=58.8 RPM, π_4=4.2 RPM, π_5=20.4 RPM, X=80, Y=350)



[Fig. 11] Number of operations vs. rate of success

[Fig. 11]는 520번의 실험을 통해 얻은 20번의 수행 당 성공률을 나타낸다. 학습이 진행됨에 따라 성공률이 증가하는 경향을 보이며 최종 학습 모델을 이용한 작업 수행 시 75%의 성공률을 보인다. 본 논문에서 제시한 방법은 보상함수를 이용한 강화학습보다 동일한 성공률을 얻기까지 더 많은 실험 수를 요한다. 하지만 실험 초기 실험 실패과정에서도 학습이 되며 보상함수 설계가 필요하지 않아 실험 준비 시간이 절약된다.

5. 결 론

본 논문은 공던지기 제어 정책 예측의 심층 강화학습에 관한 연구이다. 기존의 강화학습에서 주로 사용되어온 보상함수 설계를 통한 경사 하강법을 이용한 학습 모델은 평가를 위한 함수 설계가 필요하다. 하지만 그로 인해 작업환경에 따른 범용성이 부족하였다. 이를 극복하고자 제안된 성공과 실패 평가 방식은 측정이 간편하여 직관적으로 작업을 평가할 수 있다.

공 던지기 작업의 수행마다 성공과 실패 평가가 이뤄지며 이전 데이터들을 이용해 학습이 이뤄진다. 이후 무작위 제어 정책에 대한 성공확률을 예측하고 가장 성공확률이 높은 제어 정책을 탐색하고 실행한다. 학습, 예측, 실행 및 평가를 반복함으로써 데이터가 축적되고 학습모델의 성능은 상승한다.

본 논문에서 제시한 학습 알고리즘은 보상함수의 설계 없이 인공지능망을 이용해 제어 정책의 성공 확률을 예측할 수 있다. 이를 통해 데이터 축적에 따라 점층적 성능 향상 효과를 얻을 수 있다. 또한 성공 확률 예측을 통해 이전에 시도하지 않은 작업에 대한 예측이 가능하여 새로운 작업에 대해 지속적이고 범용적으로 사용가능한 학습모델이다.

520번의 공 던지기 수행하면서 작업의 성공률은 증가하는 경향을 보인다. 학습 결과 75%의 성공률을 보장하는 학습모델을 얻을 수 있었다.

본 논문에서 사용한 학습 알고리즘은 공 던지기와 같은 성공과 실패가 명확한 작업의 경우 손쉽게 적용가능 하다. 하지만 경로 이동, 연속적인 동작이 이뤄지는 작업의 경우 성공과 실패 평가에 어려움이 있다. 향후 연구 방향은 이러한 연속적이고 복잡한 로봇 강화학습에 적용할 수 있는 학습 알고리즘을 개발하고 일반화하는 것이다.

References

[1] U. Jung, J. M. Park, S. H. Yu, H. Lee, S. Ahn, and Y. Lee, "Design of an Efficient Robotic Arm Model for Accurate Throwing," *The Korean Association of Computer Education*, vol. 19, no. 2, pp. 121-125, 2015, UCI(KEPA): I410-ECN-0101-2018-037-003283830.

[2] E. W. Aboaf, C. G. Atkeson, and D. J. Reinkensmeyer, “Task-Level Robot Learning,” *1988 IEEE International Conference on Robotics and Automation*, Philadelphia, PA, USA, 1988, DOI: 10.1109/ROBOT.1988.12245.

[3] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, “Tossingbot: Learning to Throw Arbitrary Objects with Residual Physics,” *IEEE Transactions on Robotics*, vol. 36, no. 4, Aug., 2020, DOI: 10.1109/TRO.2020.2988642.

[4] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, “Deep Predictive Policy Training using Reinforcement Learning,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, 2017, DOI: 10.1109/IROS.2017.8206046.

[5] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement Learning in Robotics: A Survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, 2013, DOI: 10.1177/0278364913495721.

[6] E. Bıyık and D. Sadigh. “Batch Active Preference-Based Learning of Reward Functions,” *arXiv:1810. 04303*, 2018, [Online], https://arxiv.org/abs/1810.043 03.

[7] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, “Reinforcement Learning to Adjust Robot Movements to New Situations,” *Autonomous Robots*, vol. 33, 2011, DOI: 10.1007/s10514-012-9290-3.

[8] J. Y. Park, G. B. Jeong, and Y. J. Moon, “Performance Comparison of Crawling Robots Trained by Reinforcement Learning Methods,” *Korean Institute of Intelligent Systems*, vol. 17, no. 1, pp. 33-36, 2007, http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE01019878.

[9] Y.-H, Yang, S.-H. Lee, and C.-S. Lee, “Designing an Efficient Reward Function for Robot Reinforcement Learning of The Water Bottle Flipping Task,” *The Journal of Korea Robotics Society*, vol. 14, no. 2, June, 2019, DOI: 10.7746/jkros.2019.14.2.081.

[10] P. Abbeel, M. Quigley, and A. Y. Ng, “Using Inaccurate Models in Reinforcement Learning,” *The 23rd international conference on Machine learning*, 2006, DOI: 10.1145/1143844.1143845.

[11] H. Yokota, S. Ohshima, and N. Mizuno, “Information visualisation of Optimised Underhand Throw for Cybernetic Training,” *Procedia Engineering*, vol. 112, 2015, DOI: 10.1016/j.proeng.2015.07.239.

[12] J. Kober and J. Peters, “Policy Search for Motor Primitives in Robotics,” *Springer Tracts in Advanced Robotics*, vol. 97, 2009, DOI: 10.1007/978-3-319-03194-1_4.

[13] S. Jung and T. C. Hsia, “A New Neural Network Control Technique for Robot Manipulators,” *1995 American Control Conference-ACC’95*, Seattle, WA, USA, 1995, DOI: 10.1109/ACC.1995.529374.

강 영 균

2016 서강대학교 기계공학과(학사)

2020 서강대학교 기계공학과(석사과정)

관심분야: Robotics, Deep Reinforcement Learning

이 철 수

1990 KAIST 산업공학과 박사

현재 서강대학교 기계공학과 교수

관심분야: Robot & Automation Systems, Deep Reinforcement Learning