Journal of Korea Robotics Society (2019) 14(1):040-049
https://doi.org/10.7746/jkros.2019.14.1.040 ISSN: 1975-6291 / eISSN: 2287-3961

백스터 로봇의 시각기반 로봇 팔 조작 딥러닝을 위한 강화학습 알고리즘 구현

Implementation of End-to-End Training of Deep Visuomotor Policies for Manipulation of a Robotic Arm of Baxter Research Robot

김 성 운[1] · 김 솔 아[2] · 하파엘 리마[3] · 최 재 식†
Seongun Kim[1], Sol A Kim[2], Rafael de Lima[3], Jaesik Choi†

Abstract: Reinforcement learning has been applied to various problems in robotics. However, it was still hard to train complex robotic manipulation tasks since there is a few models which can be applicable to general tasks. Such general models require a lot of training episodes. In these reasons, deep neural networks which have shown to be good function approximators have not been actively used for robot manipulation task. Recently, some of these challenges are solved by a set of methods, such as Guided Policy Search, which guide or limit search directions while training of a deep neural network based policy model. These frameworks are already applied to a humanoid robot, PR2. However, in robotics, it is not trivial to adjust existing algorithms designed for one robot to another robot. In this paper, we present our implementation of Guided Policy Search to the robotic arms of the Baxter Research Robot. To meet the goals and needs of the project, we build on an existing implementation of Baxter Agent class for the Guided Policy Search algorithm code using the built-in Python interface. This work is expected to play an important role in popularizing robot manipulation reinforcement learning methods on cost-effective robot platforms.

Keywords: Robotics, Visuomotor Policy, Reinforcement Learning, Guided Policy Search, Baxter Research Robot

1. 서 론

강화학습은 주어진 환경안에서 정의된 에이전트가 현재의 상태를 인식하고, 해당 상태에서 보상을 최대화하는 행동을 선택하는 방법이다. 보상을 얻기 위해서는 시행착오를 거치는 과정(exploration - exploitation)이 필요하고, 이 트레이드오프를 감수하며 얼마나 적은 샘플과 얼마나 빠른 시간안에 좋은 정책을 학습할 수 있느냐가 관건이다. 최근 게임 분야에서 인간의 능력을 뛰어넘는 성과를 보여주었으며[1,2], 컬링과 같은 연속적 공간을 탐색하고 학습해야 하는 게임의 경우에서도 마찬가지였다[3]. 그 외 여러 방면(게임, 금융, 자율주행차, 지능형 로봇 등)에서 적용기술에 대한 연구가 활발히 진행되고 있다[4,5].

지능형 로봇 에이전트에 대한 강화학습의 적용은 특히 많은 어려움과 제약에 직면하게 된다[5]. 로봇이라는 에이전트의 상태와 행동은 본질적으로 연속공간에 있어, 각각이 나타내는 정보와 찾고자 하는 정책을 이산화 하거나 함수 추정을 통해 학습을 해야 한다.

또한, 로봇과 같은 높은 차원의 시스템은 학습과정에서 시행착오의 비용이 크기 때문에, 직접적으로 전역 정책(global policy)를 찾는 것은[6,7] 시간적 소모가 크다. 반면에, 차동 동적

Received : Dec. 13. 2018; Revised : Jan. 10. 2019; Accepted : Jan. 10. 2019
※ This project was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2017-0-00255, Autonomous digital companion framework and application) and NRF-2018M2A8A5024113.
1. Undergraduate student, UNIST, Ulsan, Korea (tjddns1597@unist.ac.kr)
2. MS/PhD student, UNIST, Ulsan, Korea (sol-a@unist.ac.kr)
3. MS graduate, UNIST, Ulsan, Korea (rafael_glima@hotmail.com)
† Associate Professor, Corresponding author: Computer Engineering, UNIST, Ulsan, Korea (jaesik@unist.ac.kr)

프로그래밍(DDP: Differential Dynamic Programming)과 같이 잘 알려진 알고리즘으로 최적제어문제를 푸는 것은 상대적으로 쉽고 빠르다.

Guided Policy Search (GPS)[8]에서는 DDP를 풀어 최적화된 국지 정책(local policy)를 사용하여 전역 정책(global policy)를 지도학습의 형태로 학습을 시킨다. 그 과정에서 KL-divergence 제약을 두어, 이전 모델과 업데이트 된 모델의 차이에 제약이 생김으로써, 전역 정책 학습의 속도뿐만이 아니라 안정성 또한 높이는 결과를 보여주었다.

해당 논문의 코드는 오픈소스로 공개되어 있다[9]. 하지만, 연구환경, 시스템, 에이전트(로봇)에 따라 바꿔야 하는 가치함수, 오프셋 등 파라미터의 수가 무수히 많아, 본 저자가 아닌 다른 연구자의 해당 실험을 재현하고 본인의 시스템에 적용하는 것은 쉽지 않다[10].

본 논문에서는 오픈소스 코드[9]의 백스터 에이전트 클래스를 Python 인터페이스에서 구현하여, 백스터 연구 로봇에서의 GPS 알고리즘이 다른 연구환경에서 적용될 수 있도록 하였다. 또한 이를 이용해 블록 조립 및 파지실험을 하여 학습과정을 분석하였다.

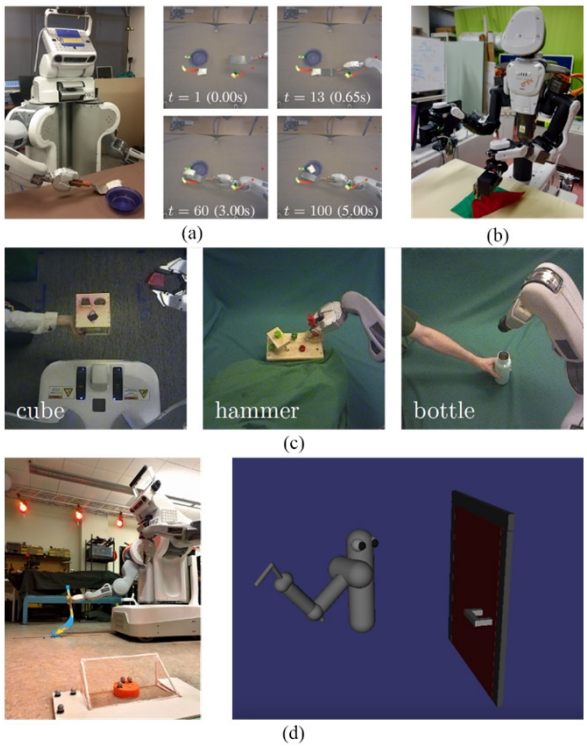
2. 관련 연구

지능형 로봇에 대한 강화학습은 이미지처리에 대한 연구와 병행되어왔다. 주로 깊은 신경망을 통해 이미지의 특징점을 추출하고 학습에 이용하는 순서로 진행된다. 본 섹션에서는 위와 같은 연구에 관련된 예시들을 소개하고자 한다. [Fig. 1]은 해당연구들이 적용된 실험을 요약한 그림이다.

오토인코더기법에서 착안하여 이미지로부터 특징점을 추출하였고, 추출된 결과를 컨트롤러의 학습에 사용한 사례가 있었다[11]. [Fig. 1(a)]와 같이 이미지로부터 추출된 특징점들이 정책을 학습하는데 필요한 상태로 변환되어 강화학습에 사용된다.

깊은 동적 정책 프로그래밍(DDPP: Deep Dynamic Policy Programming)[12]에서는 DPP에 깊은 강화학습 프레임워크를 적용하여 샘플의 효율성과 정책 강화의 안정성을 높이고자 하였다.

PILQR[13]-선형 2차 동조기-선형 적합(LQR-FLM: Linear Quadradic Regulator with Fitted Linear Models) 경로 적분 정책 향상(PI^2: Path Integral Policy Improvement) 의 결합-에서는 모델기반(model-based) 및 모델자유(model-free) 강화학습 각각의 장점을 가져와 샘플의 효율성과 수렴 후 정책의 성능을 높였다. 또한 알고리즘 자체의 우수성을 로봇실험 외에도, 시뮬레이터를 통한 다른 환경 [Fig. 1(d)]에서도 적용하여 입증하였다.



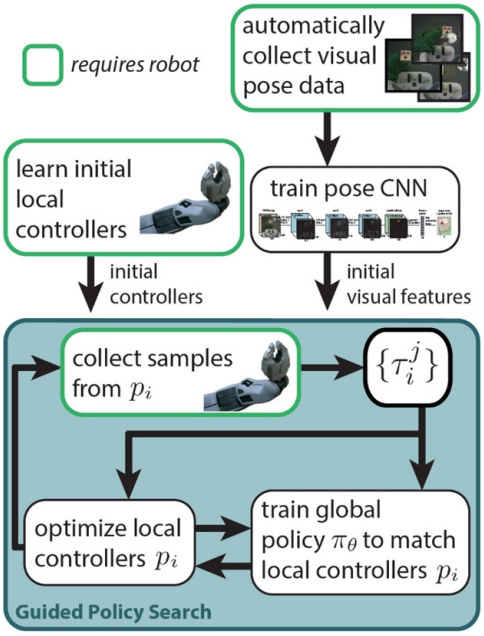
[Fig. 1] Various robotic arm manipulation tasks. (a) PR2 robot scoops a bag of rice into a bowl with a spatula[11]. (b) NEXTAGE humanoid robot flips a handkerchief[12]. (c) PR2 works on chores (inserting a cube, fitting the claw of a hammer under a nail, and screwing a cap onto a bottle)[14]. (d) PR2 robot plays hockey (left) and a simulated agent (MuJoCo) opens a door (right)[13]

3. 심층 시각기반 행동 정책의 제한된 탐색

본 섹션에서는 GPS[8]와 센서입력기반 행동 제어를 위한 GPS의 변형[14]에 대해 다룰 것이다. 영상을 입력 받아 모터를 제어하는 전역 정책을 학습하기 위해 [Fig. 2]에서 보는 바와 같이 합성곱 신경망(CNN)이 추가되었다. 또한 국지 제어기 p_i와 전역 정책 π_θ의 정합성을 가정으로 기대 비용 최소화(expected cost minimization) 문제를 풀기 위해 이중 하강 방식(dual descent method)인 브레그만 교번 방향 승수 방법(BADMM: Bregman Alternating Direction Method of Multipliers)[15]을 사용했다.

3.1 GPS

GPS 알고리즘은 다음과 같다. 본 설명은 구현중심으로 서술하기 위해 기존의 GPS의 복잡한 과정을 C-단계와 S-단계로 단순화한 Mirror Descent GPS (MDGPS)[16]로 대신한다. 1) DDP(예, 반복 LQR) 로 국지 정책 p_i를 초기화한다. 2) p_i로부



[Fig. 2] Diagram of GPS for training end to end deep visuomotor policies[14]

터 샘플 $\{\tau_{i,j}$: i번째 궤적의 j번째 샘플$\}$을 생성한다. 3) $\tau_{i,j}$로부터 선형 가우시안 역학 p_i와 선형화된 전역 정책 $\bar{\pi}_{\theta_i}$를 조정한다. 4) C-step: KL-divergence 제약과 최소 예상 비용 문제를 만족시키는 최적 제어기를 선택한다.

$$p_i \leftarrow arg\min_{\theta} E_{p_i(\tau)}\left[\Sigma_{t=1}^{T} l(\mathbf{x}_t, \mathbf{u}_t)\right] \tag{1}$$

$$D_{KL}\left(p_i(\tau) \parallel \bar{\pi}_{\theta_i}(\tau)\right) \leq \epsilon \tag{2}$$

식 (1)은 식 (2)를 만족하는 p_i에 대해서 성립한다.

5) S-단계: C-단계의 결과로부터 2)에서의 최적 입력을 계산한후, 전역 정책에 학습시킨다.

$$\pi_\theta \leftarrow arg\min_\theta \Sigma_{t,i,j} D_{KL}\left(\pi_\theta(\mathbf{u}_t|\mathbf{x}_{t,i,j}) \parallel p_i(\mathbf{u}_t|\mathbf{x}_{t,i,j})\right) \tag{3}$$

식 (3)에서와 식 (2)에서 KL-divergence값을 계산함에 있어, 두 정책의 위치가 반대임에 유의하여야 한다.

6) 2)-5)과정을 반복한다.

3.2 GPS with BADMM

BADMM은 ADMM의 변형으로 KL-divergence와 같은 비선형 제약을 갖고 있는 문제를 각각의 primal variables에 대한 Lagrangian를 번갈아 최소화하고 각각의 subgradient로 Lagrange 승수를 증가시키는 방법이다.

$$\min_{p,\pi_\theta} E_p\left[l(\tau)\right] \tag{4}$$

$$p(\mathbf{u}_t|\mathbf{x}_t) = \pi_\theta(\mathbf{u}_t|\mathbf{x}_t)\ \forall \mathbf{x}_t, \mathbf{u}_t, t \tag{5}$$

BADMM은 제약 조건, 즉 식 (5)를 만족하는 조건하에서 식 (4)에 대해 최적화된 솔루션에 수렴한다[15].

3.3 시각기반 딥러닝 정책학습

로봇의 상태 $\mathbf{x}_t$와 영상으로 부터 얻을 수 있는 정보 $\mathbf{o}_t$는 $\mathbf{x}_t \in \mathbf{o}_t$의 관계를 가지고 있다. 최종적으로 본 연구에서 학습하고 싶은 전역 정책은 $\pi_\theta(\mathbf{u}_t|\mathbf{x}_t)$가 아닌 $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$이다.

$$\pi_\theta(\mathbf{u}_t|\mathbf{x}_t) = \int \pi_\theta(\mathbf{u}_t|\mathbf{o}_t)p(\mathbf{o}_t|\mathbf{x}_t)d\mathbf{o}_t \tag{6}$$

BADMM은 같은 상태 분포 $\mathbf{x}_t$를 갖는 두 정책에 대해 수렴성을 보장하므로 식 (6)과 같이 바꾸어 학습한다. 하지만 두 정책의 등제한조건을 만족하도록 학습이 진행됨에 따라 $p(\mathbf{o}_t|\mathbf{x}_t)$에 대한 정보는 필요하지 않게 된다.

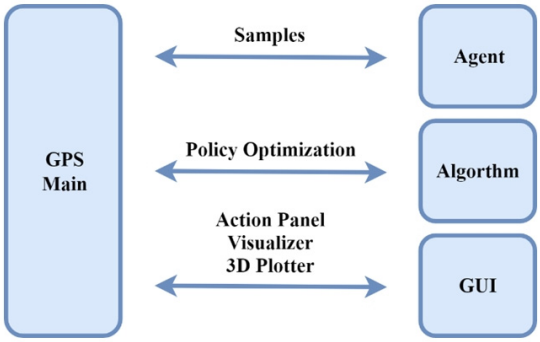
4. GPS를 백스터 연구 로봇에 적용하는 과정

본 장에서는 GPS 알고리즘의 백스터 연구 로봇으로의 구현에 대한 내용을 다룬다. GPS 알고리즘을 개발한 버클리대학에서 코드를 공개했지만[9], 공개된 코드는 Box2D 시뮬레이터, MuJoCo 시뮬레이터, 그리고 PR2 로봇을 대상으로 작성되었다. 이는 물리적으로 존재하는 로봇 중에는 PR2 이외에는 작동하지 않음을 의미한다. 따라서 본 장에서는 이 알고리즘이 어떻게 다른 로봇에 적용되었는지, 특히 이 경우, 협업 로봇 플랫폼으로 널리 사용되고 있는 백스터 연구 로봇에 적용되었는지에 대해 다룬다.

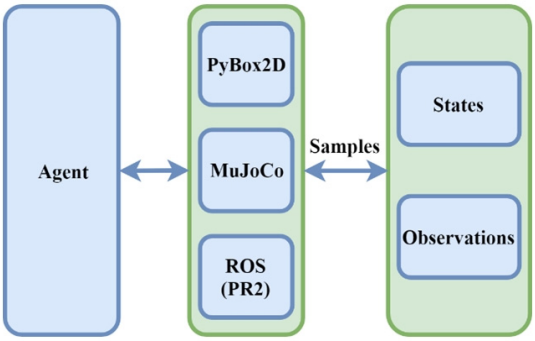
4.1 Guided policy search 코드의 구조

기존의 GPS 코드는 [Fig. 3]과 같이 에이전트, 알고리즘 및 GUI의 세 개의 주요 모듈과 함께 GPS 메인 클래스로 구성되어 있다.

GPSMain은 메인 클래스의 코드로써 다음과 같은 기능들을 수행한다. 첫번째로 에이전트 클래스와 소통하며 현재 정책으로부터의 궤적 샘플링을 수행한다. 두번째로는 알고리즘 클래스, 예를 들어, BADMM 알고리즘[15]이나 Mirror Descent 알고



[Fig. 3] The structure of the GPS code



[Fig. 4] Three kinds of GPS agent class: PyBox2D, MuJoCo, ROS-PR2

리즘[16]과 소통하며 궤적 최적화를 진행하고 신경망을 업데이트한다. 마지막으로 GUI 클래스와 소통하며 인터페이스 관리의 역할을 수행한다.

메인 클래스의 첫번째 기능인 에이전트 클래스와의 소통을 통한 샘플링은 [Fig. 4]와 같이 자세히 설명될 수 있다. 앞서 설명했듯이, 현재 GPS 코드는 세개의 에이전트 위에서 작동하는데 [Fig. 4]가 이를 잘 나타낸다. 각각의 에이전트 클래스는 해당되는 에이전트를 제어하며 매 타임스텝에서의 상태와 관찰값을 저장한다. 따라서 백스터 연구 로봇을 GPS 알고리즘 위에서 작동하게 하기 위해서는 백스터 로봇을 제어하기 위한 에이전트 클래스의 구현이 필요하다. 이에, 다음 장에서는 백스터 에이전트 클래스 설계에 대한 내용을 다룬다.

4.2 백스터 에이전트 클래스의 설계

백스터 연구 로봇을 이 알고리즘 위에서 작동시키기 위해서는 해당되는 에이전트 클래스의 구현이 필요하다. 백스터 에이전트 클래스는 두 가지 방법으로 구현될 수 있다. 첫번째는 기존에 구현되어 있는 PR2를 위한 ROS 에이전트를 기반으로 하여 백스터 에이전트 클래스를 구현하는 것이고, 두번째는 MuJoCo 에이전트처럼 Python 커맨드를 이용하여 토크 명령을 직접 주도록 구현하는 것이다.

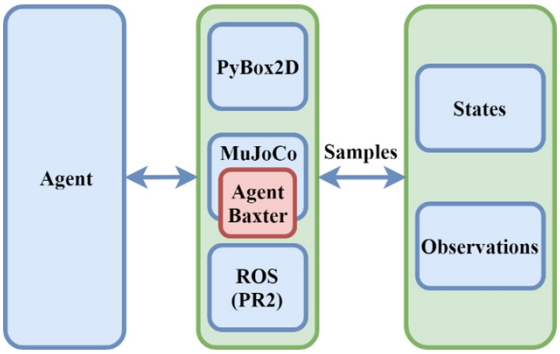
4.2.1 ROS 에이전트 기반 백스터 에이전트 설계

ROS 기반 에이전트 위에 백스터 에이전트를 구현하기 위해서는 백스터 로봇의 플러그인인 'baxterplugin' 클래스를 PR2 ROS 구현의 플러그인인 'robotplugin' 클래스로 서브 클래스화 해야 한다. 하지만 PR2 로봇의 플러그인과 백스터 로봇의 플러그인의 다음과 같은 차이로 인해 쉽지 않은 작업이다. 예를들어 PR2 로봇에는 정방향 기구학 계산과 같은 C++ API가 있는 반면, 백스터 로봇에는 C++ API가 없었다. 문제를 해결하기 위해서는 PR2 로봇의 코드 중 C++로 구현되어 있는 모든 부분을 백스터 로봇의 C++ 코드로 다시 구현해주는 것이었다. 정방향 기구학 계산과 같이 가장 낮은 단계의 코드부터 모든 C++ 코드를 다시 구현해주는 것은 현실적으로 어려운 점이 많았다. 또한 기존의 PR2 로봇을 위한 ROS 메시지들과 백스터 로봇의 ROS 메시지에도 호환이 되지 않는 부분이 있었다.

4.2.2 MuJoCo 에이전트 기반의 백스터 에이전트 설계

ROS 에이전트 기반 백스터 에이전트 클래스의 설계는 ROS 메시지, 플러그인 및 매개 변수 조정 등이 기존 ROS 에이전트와 호환이 되지 않는 문제로 백스터 에이전트를 직접 ROS 에이전트의 서브클래스로 구현하는 것에 한계가 있었다. 그 대안으로, MuJoCo 에이전트와 같 이 매 시간 스텝마다 로봇에 직접 토크 명령을 주는 방법이 있었다. 즉, ROS 에이전트처럼 알고리즘에서 온보드로 ROS 통신을 통해 로봇을 제어하는 것이 아닌, 오프보드로 외부 클래스에서 ROS 통신을 하여 로봇을 제어하는 것이다.

이를 위해서, 알고리즘 외부에서 백스터 연구 로봇을 제어해줄 BaxterMethods 클래스를 구현하였다. 이 클래스 에서는 백스터 연구 로봇의 제조사인 Rethink Robotics에서 제공하는 로봇 제어를 위한 Python API를 이용하여 로봇을 관절 제어하는 함수와 로봇의 상태, 관측값 등을 가져오는 샘플 추출을 위한 함수들이 구현되었다. [Fig. 5]와 같이 백스터 에이전트를 MuJoCo 에이전트 클래스를 기반으로 구현함으로써, 백스터 연구 로봇이 기존의 GPS 알고리즘 코드 위에서 작동할 수 있었다.



[Fig. 5] Implemented Baxter agent class under MuJoCo agent class

5. 실 험

이전 섹션에서 설명했듯이, GPS 프레임워크를 기반으로 한 End-to-End Training of Deep Visuomotor Policy[14]는 병 뚜껑을 닫고, 블록을 조립하고, 옷걸이를 거는 등 실생활 문제를 해결하는데 좋은 모습을 보였다.

이 알고리즘은 로봇의 카메라로 들어오는 이미지를 통해 직접 로봇의 관절을 제어하여 로봇이 태스크를 수행하도록 학습시킨다. 일반적으로 이미지 입력을 사용하여 관절을 직접 제어할 경우 입력의 차원이 높기 때문에, 특히 로봇의 경우 하나의 궤적에 많은 이미지가 들어가기 때문에 학습이 수렴하는데 많은 시간이 걸리고 지역 최적에 빠질 확률도 높다. 하지만 이 알고리즘에서는 최적 제어 문제를 풀어 궤적 최적화를 함으로써 학습이 빠르게 수렴하게 하고 지역 최적에 빠질 확률도 줄여준다.

본 연구에서는 이 알고리즘이 백스터 연구 로봇에 구현되었으며 블록 조립 작업과 블록 파지 작업의 실험을 통해 위와 같은 장점들이 확인되었고 태스크를 수행하는데 있어서 생길 수 있는 기술적 문제들이 검토되었다.

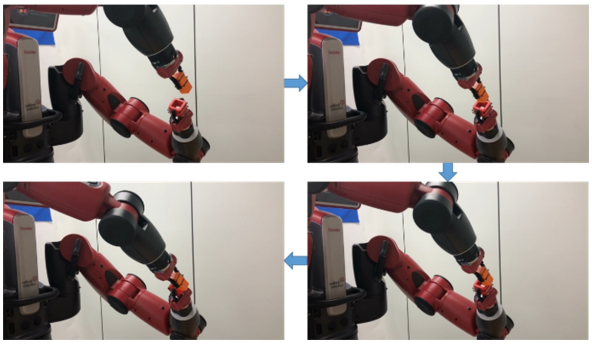
5.1 블록 조립 태스크

블록 조립 작업의 목표는 [Fig. 6]과 같이 사각형 모양의 블록을 완전히 꼭 들어맞는 구멍에 삽입하는 것이다. 처음에는 한 궤적의 매 시간스텝마다 로봇의 말단 장치의 위치와 목표 위치 사이의 차이 값과 로봇의 관절에 적용된 토크 값이 손실 함수로 사용되었다. 말단 장치의 위치와 목표 위치 사이의 차이 값은 마지막 스텝에서 로봇 팔의 말단 장치가 목표 위치에 위치하도록 하기 위해 사용되었고, 관절 토크는 최소한의 노력으로 목표 위치에 도달하도록 하기 위해 사용되었다.

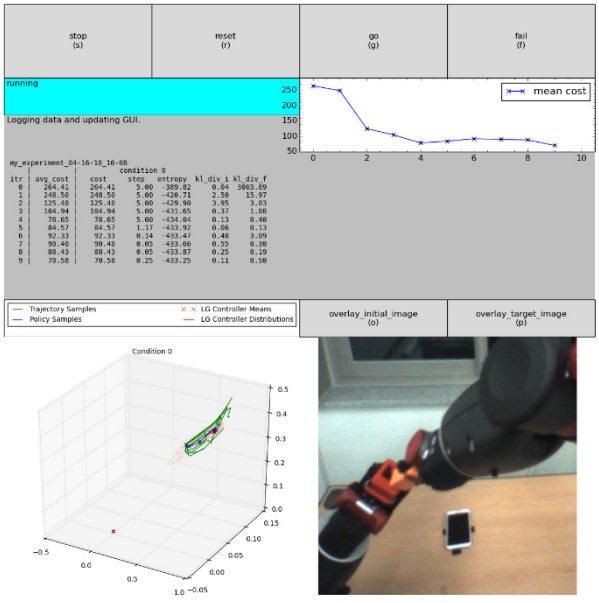
위의 손실 함수를 사용하는 것은 합리적으로 보였다. 하지만 충분히 많은 수의 궤적 최적화된 샘플이 전역 정책의 학습에 사용되었으나 블록을 조립하는데 성공하지 못하였다. 심지어 궤적의 최종 스텝에서 블록의 위치는 삽입해야 할 구멍으로부터 많이 떨어져 있었다. 결과가 좋지 않은 이유는 로봇 팔의 높은 자유도 때문이었다. 7 자유도의 로봇 팔에는 오리엔테이션이 존재하기 때문에 목표 위치에 도달하는 경우의 수가 거의 무한하다. 하지만 말단 장치의 위치만으로는 오리엔테이션에 대한 정보를 포함하지 못하기 때문에, 이를 손실 함수로 사용하는 것만으로는 학습이 전역 최적에 수렴하지 못했다.

위와 같은 문제를 해결하기 위해 말단 장치 위치의 차이가 손실 함수에 이용되는 대신, 오리엔테이션에 대한 정보를 포함하고 있는 관절 위치가 손실 함수로 사용되었다. 목표 관절 각도와 현 스텝에서 관절 각도의 차이가 손실 함수로 사용된 후, 빠른 속도로 학습이 가능하였다. 로봇의 오른손에 있는 블록이 10번째 학습반복 이후(이 실험의 경우, 한 번의 학습에 10개의 최적 궤도 작업이 사용되었기에, 즉 100번의 학습 작업 이후) 왼손에 있는 구멍과 충돌하기 시작했다. [Fig. 7]의 로봇 카메라에 찍힌 이미지는 구멍과 충돌하기 시작한 블록을 보여준다. 손실 함수를 바꾼 후 이전보다 좋은 결과를 보이며 목적 위치에 더 가까워졌지만, 학습이 전역 최적에 수렴되지는 못했다. 이는 여전히 전역 정책의 탐색 공간이 너무 크기 때문이었다.

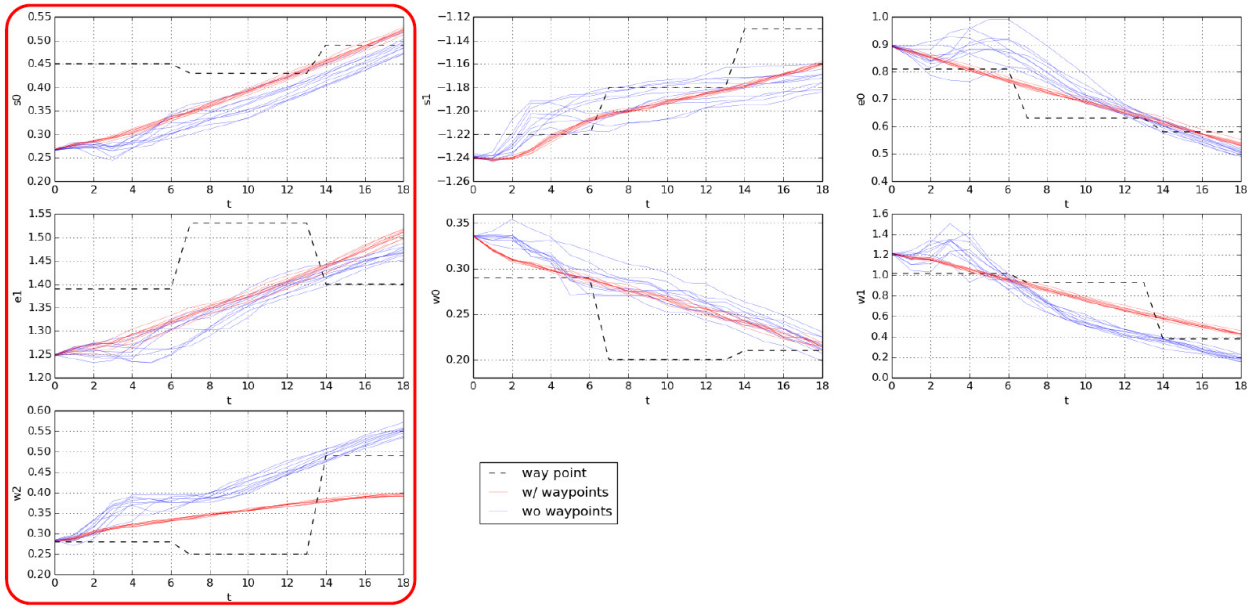
정책의 탐색 공간을 줄이고 지역 최적을 벗어나게 만들기 위한 새로운 손실 함수로써 목표 중간 지점(way points)이 설정되었다. 즉, 궤적의 전체 스텝을 부분 스텝으로 나누고, 각 부분 스텝마다 중간 목표 지점을 설정함으로써 목표 위치와 현 스텝에서 로봇 팔의 위치 차이가 새로운 손실 함수로 사용되었다. 특히 이 경우에는, 로봇 팔의 말단 장치가 아닌 관절 각도와 중간 목표 위치와의 차이가 사용되었다. 이를 통해, 하나의 긴 궤적은 여러 개의 짧은 궤적으로 쪼개진 것과 같은 효



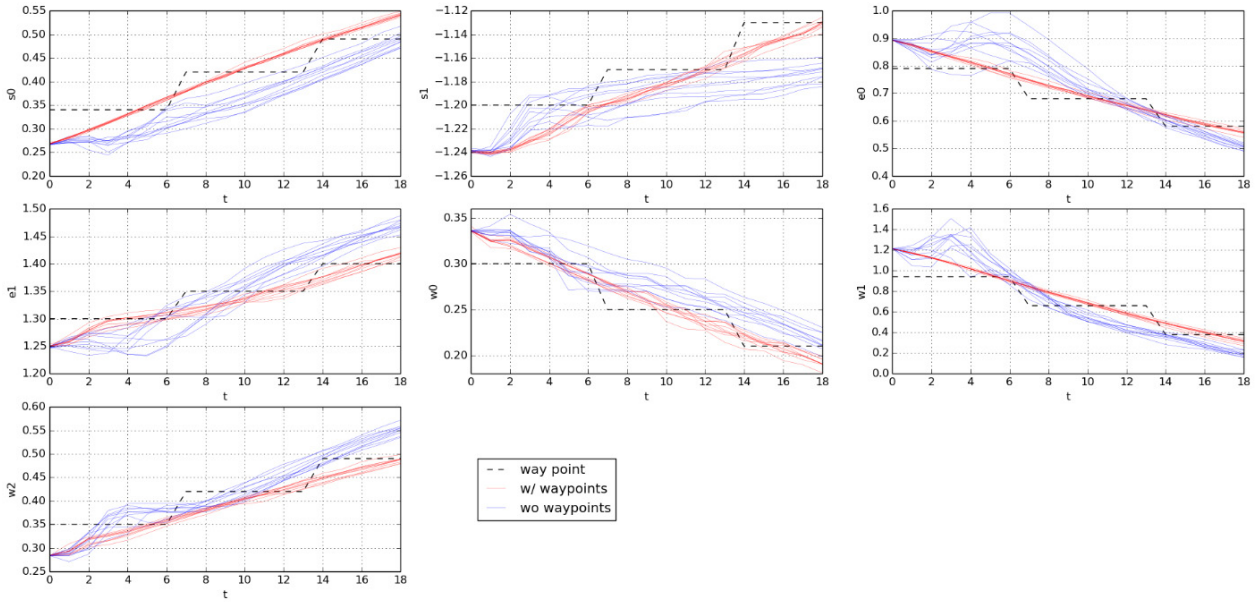
[Fig. 6] Example of the block inserting task



[Fig. 7] Cost graph when using the difference of the joint positions as a cost and the RGB image from the robot's head camera



[Fig. 8] Position of each joints (s0, s1, e0, e1, w0, w1, w2) at each time step. Red line and blue line are joint positions when way point is randomly given and not given respectively. Black dotted line is the target way points for each sub step



[Fig. 9] Position of each joints (s0, s1, e0, e1, w0, w1, w2) at each time step. Red line and blue line are joint positions when way point is linearly given and not given respectively. Black dotted line is the target way points for each sub step

과를 볼 수 있었고, 지역 최적을 벗어나는 효과를 보였다. 이는 또한 블록 조립과 같은 로봇의 정밀한 제어에도 효과적이었다.

중간 지점을 손실 함수로 사용하는 것에 대한 성능 향상을 검증하기 위해, 최종 정책이 실행되는 동안 중간 지점 손실 함수를 사용한 경우와 그렇지 않은 경우에 대해 한 궤적에서의 관절 위치를 그래프로 표현하였다. 중 간 지점 손실 함수를 사용한 경우, 전체 19 스텝이 세 개의 부분 스텝으로 나뉘었다. 첫번째 부분 스텝에서는 7 스텝, 두번째 부분 스텝에서도 7 스텝, 그리고 나머지 부분 스텝은 5스텝으로 설정되었고, 각 부분 스텝마다 목표하는 중간 지점이 다르게 설정되었다. 중간 지점 손실 함수를 사용하지 않은 경우에서는 최종 위치만이 목표 지점으로 사용되었다. [Fig. 8]은 7 자유도를 가진 백스터 연구 로봇에서 최종 전역 정책을 10회 실행하였을 때의 그래프를 보여준다. 각 그래프는 순서대로 s0, s1, e0, e1, w0, w1, 그리고 w2 관절의 시간에 따른 위치를 표현한 그래프이며, s 관절은 어깨, e 관절은 팔꿈치, w 관절은 손목에 해당한다.

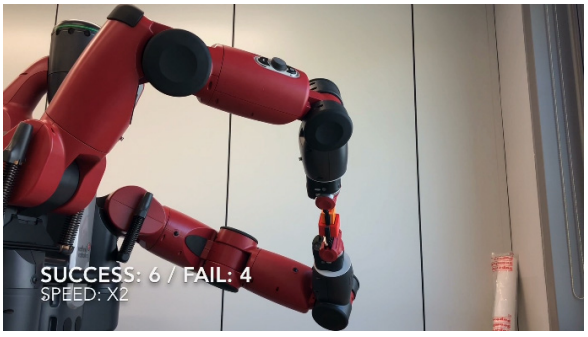
그림에서 나타나 있듯, 대부분의 경우 중간 지점 손실 함수를 사용한 후 관절 위치가 목표 위치에 더 안정적이고 정확하게 수렴하는 것을 확인할 수 있다. 하지만 그림에서 빨간 박스 안에 표시된 3개의 관절, 's0', 'e1', 그리고 'w2'에 대해서는 그렇지 않았다. 이 관절들에서는 중간 지점 손실 함수를 사용하지 않은 경우가 최종 목표 위치에 더 가까웠다.

몇 관절들에 대해, 중간 지점 손실 함수를 사용한 경우에서의 성능이 나빴던 이유는 목표 위치의 비선형성 때문으로 생각된다. 중간 지점 손실 함수를 사용했을 때 성능이 더 나빴던 관절들, 즉 관절 's0', 'e1', 그리고 'w2'에서 관절의 중간 목표 지점들은 선형적이게 증가하거나 감소하게 설정되지 않았다. 예를 들어 [Fig. 8]의 첫번째 그래프에 나타나 있는 관절 's0'의 경우, 첫번째 목표 중간 지점의 각도보다 두번째 각도가 더 크게 설정됐지만 최종 목표 지점의 각도는 두번째 각도보다 더 작게 설정되었다. GPS 프레임워크에서 최적 궤적 샘플을 만들기 위해서는 DDP, 특히 LQR 알고리즘을 사용해야한다. 그러기 위해서는 알려지지 않은 로봇의 역학이 선형으로 도출되어야한다. 하나의 궤적에서 중간 목표 지점이 높아졌다가 낮아지거나, 반대로 낮아졌다가 높아지는 것은 이 알고리즘으로 하여금 로봇의 역학을 선형으로 도출하는 것을 어렵게 만든다. 이러한 이유에서, 중간 목표 지점이 선형성을 가지도록 재설정되었고, 위와 동일한 실험을 반복하였다.

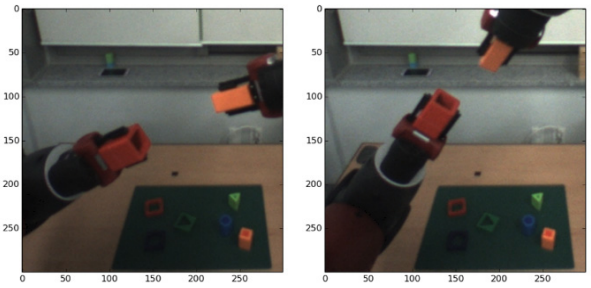
[Fig. 9]는 중간 목표 지점을 선형적으로 증가 혹은 감소하게 줬을 때의 각 관절의 위치를 나타내는 그래프이다. 여기서 확인할 수 있듯, 선형 목표 중간 지점을 손실 함수로 사용하는 것은 최종 목표 지점 하나만을 손실 함수로 사용하는 것보다 훨씬 안정적이고 최종 목표 지점에 가까이 도달할 수 있었다.

중간 목표 지점을 설정했을 때, 더 정교하고 가깝게 최종 위치에 도달할 수 있는 장점에도 불구하고, 학습은 빠르게 수렴되지 않았다. 로봇 팔이 동일한 스텝에 최종 목표 위치까지 더 가까이 도달할 수 있음에도 설정된 중간 목표 지점이 이를 막는 것이었다.

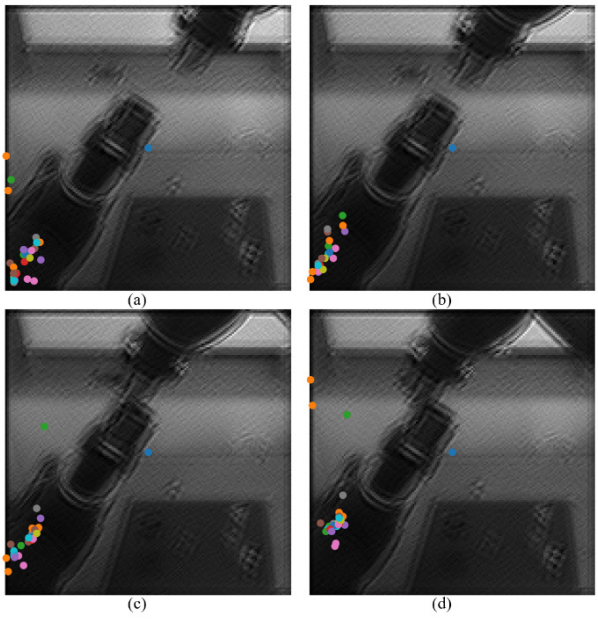
이 문제는 두 개의 손실 함수를 동시에 사용하는 것으로 간단히 해결될 수 있었다. 중간 목표 지점을 설정하는 손실 함수와 최종 목표 지점만을 설정하는 손실 함수를 같이 사용함으로써 학습이 지역 최적을 벗어날 수 있었고 더 빠르게 수렴할 수 있었다. 또한, 이를 통해 [Fig. 10]와 같이 블록 조립 태스크에 성공할 수 있었다. 이 실험에서는 총 10번의 최종 전역 정책이 실행되었는데, 60%의 성공률로 로봇 팔이 블록을 조립하는데 성공하였다. 블록 조립 태스크를 성공한 후에, 전역 정책을 위한 신경망의 공간적 소프트맥스 계층이 얼마나 잘 작동하는지 검증하기 위해 마지막 합성곱 계층과 함께 특징점을 시각화하였다. 실험은 [Fig. 11]과 같이 로봇이 블록을 수 직으로 조립하려 할 때와 수평으로 조립하려 할 때, 두 가지 경우에서 진행되었다. 이 두 가지 경우는, 공간적 소프트맥스 계층에서 유도되는 특징점이 명백히 다른 이미지에 대해 얼마나 적절하게 대응하는지를 확인하기 위해 각각의 최적 궤적 샘플을 실행한 후 전역 정책을 학습할 때는 한 번에 학습되었다.



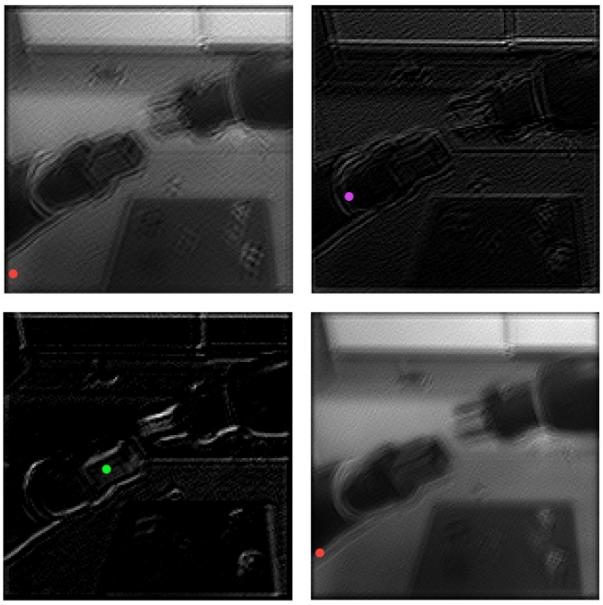
[Fig. 10] Image of the robot arm in the final time step when it succeeds to insert the block



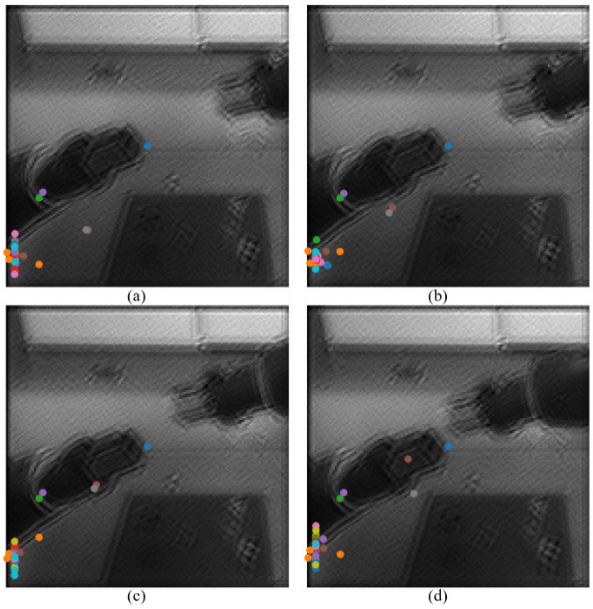
[Fig. 11] Multiple conditions with different target positions for block inserting task seen by robot's monocular camera on its head



[Fig. 12] Feature points plotted on the last convolution layer at time step (a) 1, (b) 7, (c) 13, and (d) 20 when it inserts vertically. There are total 32 feature points each of which comes from its corresponding channel for each convolution layer

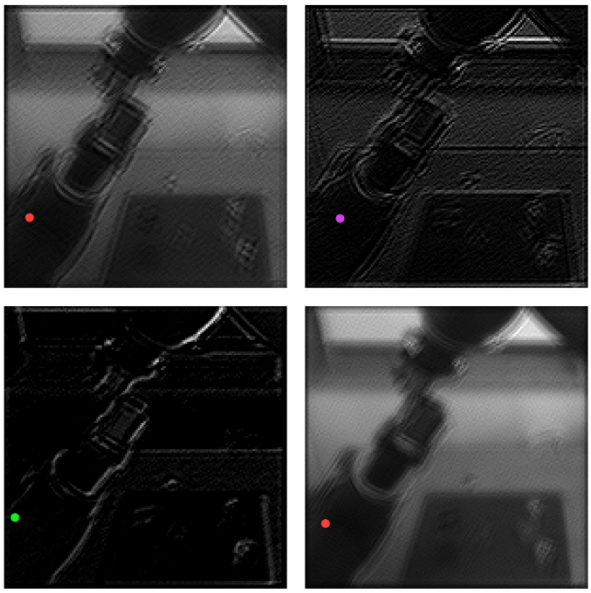


[Fig. 13] Feature points plotted on the last convolution layer for each channel at the final time step when it inserts vertically. Four channels are selectively chosen among 32 channels. Color of each points is selected randomly

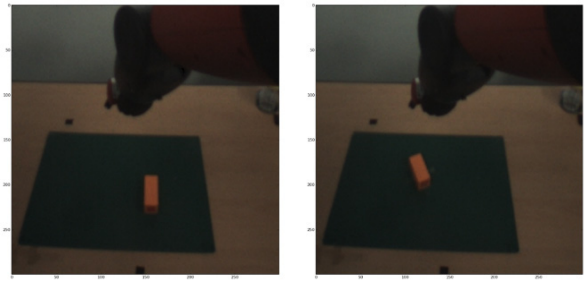


[Fig. 14] Feature points plotted on the last convolution layer at time step (a) 1, (b) 7, (c) 13, and (d) 20 when it inserts horizontally. There are total 32 feature points each of which comes from its corresponding channel for each convolution layer

[Fig. 12]와 [Fig. 13]은 로봇 팔이 블록을 수직으로 조립하려 할 때의 상황이고, 반대로 [Fig. 14]와 [Fig. 15]는 수평으로 조립하려 할 때의 상황이다. 이 실험을 통해, 비록 특징점이 이미지 내에서 의미있는 곳에 찍히지는 않았지만, 두 개의 명백히



[Fig. 15] Feature points plotted on the last convolution layer for each channel at the final time step when it inserts horizontally. Four channels are selectively chosen among 32 channels. Color of each points is selected randomly



[Fig. 16] Multiple conditions with different target position for block grasping task seen by robot's monocular camera on its head

다른 상황에 대해서 딥러닝 제어기의 내부 노드가 다르게 인식하고 있다는 것을 확인할 수 있었다.

특징점이 의미있는 곳에 찍히지 않은 한 가지 이유는 이미지에 있는 많은 장애물들 때문이다. 버클리대학에의 실험[14]에서는 그들은 움직이지 않는 반대쪽 로봇 팔을 비롯해 모든 태스크 수행과 관련 없는 물건들을 천으로 감싸 놓았다. 의도적으로 합성곱 심층 신경망이 태스크와 관련된 의미있는 특징들을 주목할 수 있도록 한 것이다.

5.2 블록 파지 태스크

블록 파지 태스크에 관해서는, 로봇이 단일 카메라로 파악하는 블록의 위치에 따라 여러 위치에 대해 잡을 수 있는 것을 목표로 하였다. 그러기 위해 일단 두 개의 블록 목표 좌표에 대

한 상황이 한 번의 학습으로 이루어졌다. 각각의 상황에 대해 파지해야 할 블록의 위치는 [Fig. 16]과 같이 위치하였다. 이 실험에서 학습이 완료된 후, 매 시간스텝마다 이미지 입력을 통해 직접 관절 토크 값으로 제어를 하는 이 로봇은 실제로 블록의 서로 다른 위치에 대해 다른 행동을 보였다. 이는 학습된 로봇이 카메라 이미지를 기반으로 행동한다는 것을 보여준다. 그러나 블록 파지 태스크의 입장에서 볼 때 결과는 좋지 않았다. 첫번째 상황에서 로봇은 파지해야 할 블록에 거의 근접하였다. 하지만 두번째 상황에 대해서는 그렇지 못했다.

좋지 못한 결과는 역학 정합에서 비롯되었다. GPS 프레임워크에서 로봇의 시스템 역학 전역정책을 학습시키기 위한 궤적 최적화를 하는데 사용된다. 따라서 이를 실제 로봇의 역학에 맞도록 도출시키는 것은 정책의 학습에 중요한 역할을 한다. 이 실험의 경우, 두번째 상황에서는 궤적 최적화된 샘플에서조차 파지해야 할 블록에 근접하지 못했다. 이는 역학 정합이 제대로 이뤄지지 않았음을 의미한다. 두 블록 사이의 거리가 멀 경우, 각 상황에 해당하는 국지 정책 궤적에 큰 차이가 나게 된다. 이는 선형성을 제약조건으로 하는 역학 정합 과정에서, 로봇의 역학이 선형적으로 정합되기 힘들게 한다. 따라서 두 블록 사이의 거리가 좁혀질 필요가 있었다. 이를 통해 확인할 수 있듯이 역학 정합에 대한 문제는 GPS 프레임워크의 잠재적 한계가 될 수 있었다.

6. 결론 및 추후과제

본 연구에서는 GPS 알고리즘을 백스터 연구 로봇에 구현하였다. 로봇은 이 알고리즘을 통하여 시각을 기반으로 블록 삽입 및 파지 작업을 수행하였다. GPS 프레임워크는 2차 동적 미분 프로그래밍(DDP)을 통해 얻어진(특히 이 논문의 경우, LQR 알고리즘을 통해 얻어진) 최적에 가까운 샘플 궤적을 활용하여 문제 해결을 위한 로봇의 최적의 움직임을 학습하는 강화학습 기반 알고리즘이다. 이 최적에 가까운 국지 정책은 학습이 지역 최적을 벗어날 수 있도록 도와주고, 전역 최적에 빠른 속도로 수렴할 수 있도록 도와준다. 이는 또한 학습이 다른 모델자유강화학습에 비해 훨씬 샘플 효율적으로 만들어준다. 이러한 샘플 효율적인 특성은 에이전트의 상태의 차원이 높은 로봇 학습에서 특히나 중요하다. 비록 국지 정책을 활용하기 위해서는 시스템 역학이 선형 가우시안 분포를 따라야 한다는 제약 조건이 있지만, 알고리즘은 신경망을 활용하여 여러 개의 국지 정책을 합침으로써 비선형성을 얻는다. 뿐만 아니라, 모터 스킬을 표현하는 전역 정책을 위해 신경망을 사용함으로써, 드라마틱하게 다른 상태를 표현하지는 못하지만 알고리즘이 같은 상태 분포로부터 다른 상태로 일반화하여 표현할 수 있다.

GPS 프레임워크를 새로운 로봇 환경에 적용하기 위해서는 오픈 소스로 공개된 코드에 새로운 에이전트 클래스를 구현할 필요가 있었다. 로봇의 경우 기존의 ROS 에이전트 위에 새로운 에이전트를 구현하는 것이 일반적인 방식이나, 백스터 연구 로봇의 경우 C++ API가 제공되지 않고 ROS 메시지 또한 PR2 로봇과 호환이 되지 않는 부분들이 있었다. 따라서 백스터 에이전트를 ROS 에이전트에 서브 클래스화하는 대신, 백스터 로봇을 제어하는 BaxterMethods 클래스를 구현하여 이와 통신하는 백스터 에이전트 클래스를 MuJoCo 에이전트 클래스에 서브 클래스화하였다.

로봇의 시각기반 블록 삽입 작업 및 파지 작업을 위해서는 손실 함수에 많은 수정이 필요했다. 정책의 탐색 공간을 줄이기 위해 말단 장치와 목표 좌표 사이의 거리를 손실 함수로 사용하는 대신 로봇의 관절 각도와 목표 관절 위치의 차이가 손실 함수로 사용되었다. 또한 지역 최적을 벗어나게 만들기 위해 목표 중간 지점이 설정되었다. 이 손실 함수들이 사용되지 않았을 때는 두 작업에서 150 샘플을 통한 학습에서도 모두 0%의 성공률을 보였다. 하지만 관절 각도 손실 함수와 중간 지점 손실 함수가 사용됨으로써 150 샘플을 통한 학습 후에는 블록 삽입 작업에서는 60%의 성공률을 보였고, 블록 파지 작업에서는 100%의 성공률을 보였다.

그러나 이 프레임워크에도 한계점은 있었다. GPS가 가지는 장점들을 위해서는 역학 정합이 어느정도 수준으로는 성공적으로 되어야 한다는 보장이 필요하기 때문이다. 위의 실험 섹션에서 보았듯이, 로봇 팔이 여러 개의 목표 지점을 가질 때 목표 관절 좌표가 서로 너무 다르면 역학 정합이 완벽히 이뤄지지 않아서 전역 최적으로의 수렴에 문제가 생길 수 있다. 이러한 이유에서, 역학 정합을 안정화하고 성능을 높이는 작업은 앞으로 개선할 필요가 있다.

이 알고리즘에서는 궤적을 하나의 에이전트만을 위해 최적화하기 때문에 로봇이 문제 해결을 위해 팔을 하나밖에 사용하지 못했다. 따라서 두 팔을 사용해야하는 복잡한 태스크를 수행하기 위해 이 알고리즘을 사용할 때 문제가 될 수 있었다. 한가지 제안된 추후 과제는 GPS 알고리즘을 멀티 에이전트 강화학습과 통합하여 양팔을 모두 사용할 수 있게 하는 것이다.

References

[1] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, “Mastering the game of go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484-489, January, 2016.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354-359, October, 2017.

[3] K. Lee, S.-A. Kim, J. Choi, S.-W. Lee, "Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling," *35th International Conference on Machine Learning (ICML)*, Stockholm, Sweden, pp. 2937-2946, 2018.

[4] K. Arulkumaran, M.P. Deisenroth, M. Brundage, and A.A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26-38, November, 2017.

[5] J. Kober and J. Peters, "Reinforcement learning in robotics: A survey," *Learning Motor Skills*, Springer, 2014, ch. 2, pp. 9-67.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, February, 2015.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971 [cs.LG]*, 2015.

[8] S. Levine and V. Koltun, "Guided policy search," *30th International Conference on Machine Learning (ICML)*, Atlanta, Georgia, USA, pp. 1-9, 2013.

[9] C. Finn, *Guided policy search*, [Online], https://github.com/cbfinn/gps, Accessed: January 14, 2019.

[10] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *arXiv:1709.06560 [cs.LG]*, 2017.

[11] C. Finn, X.Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, pp. 512-519, 2016.

[12] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep dynamic policy programming for robot control with raw images," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, pp. 1545-1550, 2017.

[13] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine, "Combining model-based and model-free updates for trajectory-centric reinforcement learning," *34th International Conference on Machine Learning (ICML)*, Sydney, Australia, pp. 703-711, 2017.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research (JMLR)*, vol. 17, no. 39, pp. 1-40, January, 2016.

[15] H. Wang and A. Banerjee, "Bregman alternating direction method of multipliers," *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, pp. 2816-2824, 2014.

[16] W. Montgomery and S. Levine, "Guided policy search via approximate mirror descent," *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, pp. 4008-4016, 2016.

김 성 운

2019 울산과학기술원 컴퓨터공학(학사 졸업 예정)

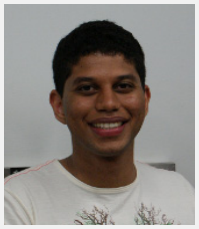
관심분야: 인공지능, 로보틱스, 강화학습, 기계학습

김 솔 아

2017 울산과학기술원 컴퓨터공학(학사)

2017~울산과학기술원 컴퓨터공학(석박사통합과정)

관심분야: 인공지능, 로보틱스, 강화학습, 기계학습



Rafael de Lima

2014 Universidade Federal da Bahia(학사)

2017 울산과학기술원 컴퓨터공학(석사)

관심분야: 필터링 알고리즘, 관계형 칼만 필터링, 기계학습

최 재 식

2004 서울대학교 컴퓨터공학과(학사)

2012 University of Illinois at Urbana-Champaign 전산학과(박사)

2013 Lawrence Berkeley National Laboratory(박사후연구원)

2013~현재 울산과학기술원 전기전자컴퓨터공학부 부교수

관심분야: 인공지능, 기계학습, 컴퓨터비전, 로보틱스