<캡스톤디자인 최종 논문>

기계학습을 이용한 그네 타기 로봇의 제어

최호승 · 박희재†
서울과학기술대학교 기계시스템디자인공학과

Control of swing robot using reinforcement learning

Ho-Seung Choi, Hee-Jae Park †
Dept. of Mechanical System Design Engineering, Seoul National Univ. of Science and Technology
† Corresponding Author, looki@seoultech.ac.kr

Keywords: Reinforcement Learning(강화학습), Actor-Critic(액터-크리틱)

초록: 본 연구는 목표는 실제 기구에 Actor-Critic 강화 학습 알고리즘을 적용하여 반복 학습을 통한 고난이도의 제어 기술을 기계학습으로 대체하는 것을 검증하는 것을 목표로 한다. 이를 위하여 비교적 간단한 동역학 모델을 가지는 1-DOF 의 그네 타기 로봇과 시뮬레이션을 제작하고, 실제 기구와 시뮬레이션을 통하여 각각 학습한 뒤 이를 비교한다. 한 개의 공유 은닉층이 액터망과 크리틱망으로 분기되는 비선형 구조의 DNN 를 설계하였고, 액터망은 에이전트가 다음에 해야 할 행동을 선택하고, 크리틱망은 행동의 좋음을 평가하는 역할을 한다. 실제 기구를 이용한 학습을 통하여 그네가 규칙적인 동작을 통하여 최대 20°까지 올라가는 swing 모션을 만들어 낼 수 있음을 확인하였다.

Abstract: The goal of this study is to apply the Actor-Critic reinforcement learning algorithm to a real machine to verify that machine learning replaces the high-level control technology through repetitive learning. To this end, a 1-DOF swing-riding robot with a simple dynamics model and a simulation model are designed. After learning, we verify performance through comparing the real machine and simulation. A DNN of a non-linear structure in which a single shared hidden layer is branched into an actor network and a critic network is designed, the actor network selects the action to be performed by the agent next, and the critic network plays a role in evaluating the goodness of the action. It was confirmed that the swing motion that rises up to 20° can be created through regular movement of the swing through the learning using the actual device.

I. 서 론

1. 연구 배경

현재 인공지능 연구는 하드웨어와 소프트웨어 기술의 비약적인 발전으로 전성기를 맞고 있다. 대부분의 인공지능 연구는 지도 학습과 비지도 학습을 이용하여 방대한 데이터를 기반으로 하여 사용자에게 편의를 제공하는 상용 소프트웨어 상품을 개발하는 것을 목적으로 한다.

한편, 강화학습을 이용한 연구는 대부분 기계 제어 분야에 집중되어 있다. 이는 기계 제어의 난이도가 높아질수록 요구하는 지식의 깊이가 기하급수적으로 깊어지기 때문이다. 따라서 기존의 제어 공학적 방법론을 기계 학습을 이용한 인공지능으로 대체하는 것이 대부분의 강화학습 목적이다.

복잡한 동역학 모델의 시뮬레이션을 제작하는 것 또한 고비용의 작업이므로 모든 환경을 시뮬레이션 할 수 없지만 대부분의 연구가 동역학 시뮬레이션 상에서 학습하는 것에 그쳐 아쉬움이 남는다.

2. 연구의 목적

본 연구의 의의는 시뮬레이션 상이 아닌 실제 기구를 사용하여 반복 학습한 인공지능 모델의 학습 가능성과 효용성을 검증하는 것에 있다.

우선 실제 기구를 제작하고 동일한 동역학 모델을 가지는 시뮬레이션을 구현한다. 그리고 각각의 환경으로 기계학습을 수행하고 학습이 끝난 모델을 교차 검증함으로써 모델의 성능을 평가한다.

3. 배경 이론

3.1 강화 학습

기계 학습은 크게 지도 학습, 비지도 학습, 강화 학습으로 세 가지로 분류된다. 지도 학습은 미리 분류된 환경과 정답을 한 쌍으로 하는 데이터를 학습시켜 새로운 데이터가 어떤 정답을 가질지 예측한다. 비지도 학습은 정답이 없는 데이터를 비슷한 특징끼리 군집화 하여 새로운 데이터의 결과를 예측한다.

강화 학습은 학습 주체가 주변 환경에 따라서 한 행동에 대하여 크고 작은 보상을 받으며 적절한 행동을 학습하여 새로운 환경에 놓였을 때, 어떤 행동이 적절한지 예측한다. 강화학습이 일반적으로 기계 제어에 많이 사용되는 이유는 강화학습은 최적 제어에 근간을 두어 데이터를 이론으로 분류할 수 없거나 정답이 없는 경우에도 학습할 수 있기 때문이다.

강화 학습에서 학습의 주체를 에이전트(Agent)라고 한다. 강화학습은 Fig.1 과 같이 에이전트는 환경(Environment, State)을 관측(Observation)하여 정책(Policy)에 따라서 다음 행동(Action)을 결정한다. 에이전트가 수행한 행동에 따라서 환경은 변화하고 이에 따라서 보상(Reward)을 받게 된다. 강화 학습의 목적은 에이전트가 어떤 환경에서 최대한 많은 보상을 얻을 수 있는 행동을 하도록 학습시키는 것이다.

최대한 많은 보상을 얻도록 행동하는 정책을 최적 정책(Optimal Policy)라고 한다. 최적 정책을 찾을 수 있는 여러 가지 방법론이 제시되었는데 대표적으로 Q-Learning, REINFORCE, Deep Q-Learning, Actor-Critic, PPO 등이 있다.

3.2 액터-크리틱(Actor-Critic, A2C) 알고리즘

액터-크리틱 알고리즘은 REINFORCE 와 Dueling DQN 알고리즘의 원리를 바탕으로 발전된 알고리즘이다. 액터-크리틱 알고리즘은 Fig.2 와 같이 Policy 가 Actor 와 Critic 으로 구성된다. Actor 는 현재 상태에서 적절하게 행동하여야 할 행동을 선택하고, Critic 은 Actor 가 수행한 행동이 적절한 행동인지 평가하여 업데이트의 기준을 만드는 역할을 한다. 이를 통하여 Agent 는 행동의 좋고 나쁨을 평가할 수 있는 Critic 을 통하여 최선을 행동을 선택하는 방향으로 학습하게 된다.

신경망 모델은 Fig.3 와 같이 액터 망과 크리틱 망으로 분리된 두 개의 신경망으로 구성된다. 액터 망은 현재 상태에서 어떤 행동을 선택해야 할지 각 행동에 대한 확률분포($\pi(s,a)$)를 출력한다. 크리틱 망은 현재 상태에서 최대로 얻을 수 있는 예측 보상(미래의 가치)을 출력하는 가치 망이다.

개념적으로는 액터 망과 크리틱 망은 분리되어 있지만, 실제 학습할 때는 학습 효율성을 위해서 공통 신경망을 거친 뒤 분기되는 비선형 망 구조로 설계한다.

3.3 행동 이득(Advantage)

특정 상태 s에서 행동 a를 하였을 때 기대되는 향후의 모든 보상의 합을 상태-행동 가치 함수, $Q(s,a)$라고 한다. 특정 상태 s 에서 최적으로 행동하였을 때 기대되는 향후의 모든 보상의 합을 상태 가치 함수($V(s)$)라고 한다.

상태-행동 가치 함수와 상태 가치 함수의 차이가 적을수록 최적으로 행동하였다는 사실을 알 수 있다. 따라서 이 차이를 행동을 얼마나 적절히 하였는지 나타내는 지표로 사용할 수 있는데, 이를 행동 이득, $A(s,a) = Q(s,a) - V(s)$이라고 한다.

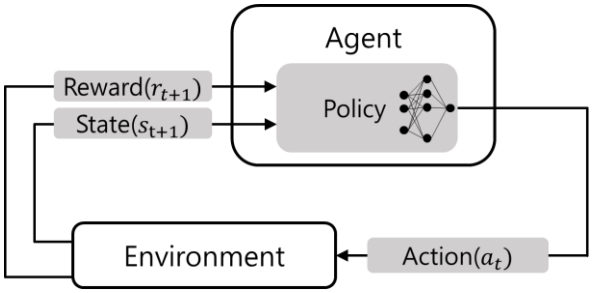

Fig. 1 Block Diagram of Reinforcement Learning

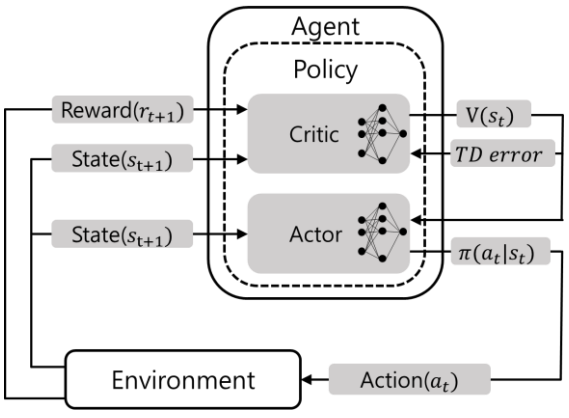

Fig. 2 Block Diagram of Actor-Critic Algorithm

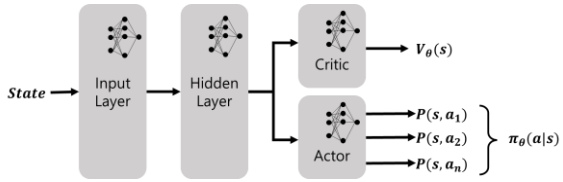

Fig. 3 Diagram of Neural Network of Actor-Critic Algorithm

3.4 Baseline

Actor-Critic 알고리즘은 REINFORCE 알고리즘에서 gradient variance 를 줄이는 방법을 통하여 수렴 안정성을 향상시킨 알고리즘이다.

강화 학습 알고리즘의 일종인 REINFORCE 의 목표 함수는 다음과 같다.

$$\nabla_\theta J(\theta) \approx E_{\pi_\theta}[Q(s,a)\nabla_\theta \log \pi(a|s)]$$

REINFORCE 목표함수의 $Q(s,a)$에 $V(s)$를 감산하여 $A(s,a)$를 사용함으로써 분산을 줄인다. $V(s)$는 $Q(s,a)$에서 빼어 분산을 줄이고 좋음과 나쁨을 판단하는 기준점이 된다. 이를 baseline 이라고 한다. 따라서 Actor-Critic 의 목표함수는 다음과 같이 수정된다. (2)

$$\nabla_\theta J(\theta) \approx E_{\pi_\theta}[A(s,a)\nabla_\theta \log \pi(a|s)]$$

이 목표함수에서 행동의 확률 분포($\pi(s,a)$) 와 행동이득($A(s,a) = Q(s,a) - V(s)$)에서 상태 가치 함수($V(s)$)를 추정하기 위하여 각각 Actor 망과 Critic 망을 사용한다.

3.5 n-step 학습

Actor-Critic 알고리즘은 매 스텝마다 학습하는 1-step 학습법, 즉 $TD(0)$에 기반하고 있다. 1-step 마다 학습하게 되면 bias 가 높은 문제가 있지만, n-step 으로 늘리고 평균을 취하여 학습하게 되면 variance 가 커지는 대신 bias 를 줄일 수 있다.

본 논문에서는 n-step 에서 단순 평균을 취하는 것이 아니라 가중된 평균값을 이용하여 계산하는 $TD(\lambda)$ 를 이용하여 Advantage 를 계산한다. 또한 Data 간 correlation 이 큰 문제를 해결하기 위하여 n 을 ∞로 계산하여 TD 보다 MC 에 가까운 알고리즘인 $TD(\lambda = 1)$를 사용하였다.

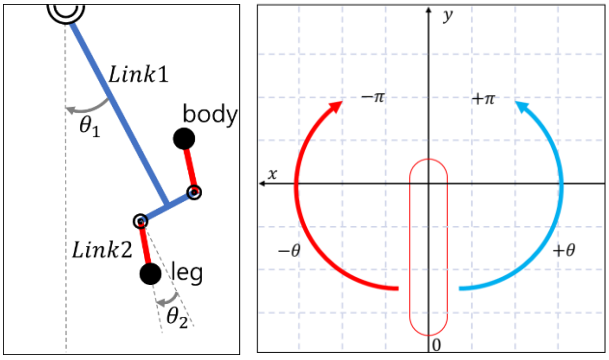

Fig. 4 Definition of Link and Joint

II. 본 론

1. 기구학적 정의

그네 타기 로봇은 상체와 하체를 각각 조종할 수 있는 2 자유도 모델로 설계하였고 본 연구에서는 하체(leg)만 사용하는 1 자유도 모델의 로봇으로 모델링 하였다. 로봇과 그네는 일체형으로 고정되어 있고, 필로우형 유닛 베어링을 통하여 360° 자유 회전이 가능하도록 설계하였다.

로봇의 기구학적 정의는 Fig. 4 와 같다. Link1 은 그네 자체, Link2 는 로봇이 가지고 있는 무게추이다. θ_1 은 중력 방향과 link2 가 이루는 각도이고, θ_2는 Link2 의 Link1 에 대한 상대적인 각도이다.

각도는 $-\pi \le \theta \le \pi$의 범위를 가지고 로봇의 좌측 옆면을 보았을 때, 반 시계 방향이 양의 부호를 가진다.

2. 학습 변수

에이전트가 학습하기 위해 필요한 주변 환경 정보인 Observation Space(관찰 공간)은 Table1 과 같다. 데이터간 correlation 을 최소화하기 위하여 두 링크의 각도와 각속도로 네 가지 파라미터만 사용하였으며, 각각 실제 기구의 최솟값과 최댓값을 측정하여 Min-Max Normalization 하였다.따라서 각 파라미터는 -1 에서 1 사이의 값을 갖는다.

에이전트가 한 State 에서 행동할 수 있는 행동을 정의한 Action Space 는 Table2 와 같다. 한 Step 에서의 Action 은 Target Angle 을 의미한다. a_{-1}의 경우 10° 의 위치로 Link2 를 이동하는 행동이다. 명령을 받으면 제어기는 설정된 모터 프로파일을 이용하여 적절한 토크를 가하여 해당위치로 이동하는 명령을 수행한다. 한 Step 의 interval 은 40ms 로 설정하였고 다음 스텝마다 새로이 행동을 결정하게 된다.

학습의 Reward 는 다음과 같이 정하였다.

$$reward = \begin{cases} |\sin(\theta_1)|, & sgn(\dot{\theta_1}) = sgn(action) \\ -|\sin(\theta_1)|, & sgn(\dot{\theta_1}) \neq sgn(action) \end{cases}$$

그네의 목표는 발산하지 않고 최대한 높이 올라가는 것이다. 기본적인 reward 는 그네의 각도가 수평에 가까울수록 큰 reward 를 받도록 구성하였다. 인간이 그네를 탈 때 그네가 움직이는 방향으로 다리를 향한다는 것에 착안하여 그네가 움직이고 있는 방향과 다리가 움직이고 있는 방향이 같은 방향이라면 양의 보상을, 방향이 반대라면 음의 보상을 받도록 구성하여 reward 와 penalty 를 줄 수 있도록 하였다.

3. 신경망 구조

본 연구에서는 Fig. 3 과 같이 입력 값이 하나의 공통 신경망을 거친 후에 각각 Actor 망과 Critic 망으로 분기되는 구조로 설계하였다.

Table. 1 과 같이 4 가지로 정의한 State 는 Input Layer 로 입력된다. Hidden Layer 에 해당하는 공통 신경망은 Fully Connected Layer 로 128 개의 node 를 가지고 ReLU 활성화 함수를 거친다. Actor 망은 Table. 2 와 같이 그네 로봇이 다리를 들어 올릴 확률과 내릴 확률을 출력한다. 확률 형태로 출력하기 위하여 가중치를 가지지 않는 SoftMax Layer 를 거치도록 되어 있다. Critic 망은 $V(s)$ 자체를 출력한다.

신경망의 모든 파라미터는 1027 개이다.

Table 1. Definition of Observation Space

1	2	3	4
θ_1	θ_2	$\dot{\theta}_1$	$\dot{\theta}_2$

Table 2. Definition of Action Space

-1	1
CCW (10°)	CW (−100°)

Table 3. Min-Max of Observation Space

Parameter	θ_1 $[rad]$	θ_2 $[rad]$	$\dot{\theta}_1$ $[rad/s]$	$\dot{\theta}_2$ $[rad/s]$
Min	-1.1972	-0.1951	-5.7322	20.8392
Max	1.2165	1.7601	5.5671	-19.1637

4. 실험 기구

전체 실험 기구는 $420 \times 510 \times 625\ (mm)$ 의 크기로 모터의 구동 시 무게추와 프레임 간의 간섭이 없도록 설계하였다.

로봇은 등에 배터리와 제어기를 내장하여 블루투스를 사용한 원격 학습이 가능하도록 설계하였다. 다리에 해당하는 무게추를 움직이는 모터는 로보티즈社의 MX-106T 를, 상체를 움직이기 위한 모터는 MX-64T 모터를 선정하였다. 무게추는 약 300g 으로 A6061 알루미늄 재질을 사용하였다.

전체 시스템은 Fig. 7 과 같이 Desktop 에서 Actor-Critic 인공 신경망이 추론한 행동 값을 UART 통신을 통하여 OpenCM9.04 제어기로 전달하고 제어기는 TTL 통신을 사용하여 각 모터를 제어하고 데이터를 수집한다. 수집한 데이터는 제어기가 다시 Desktop 으로 UART 통신을 이용하여 전송한다.

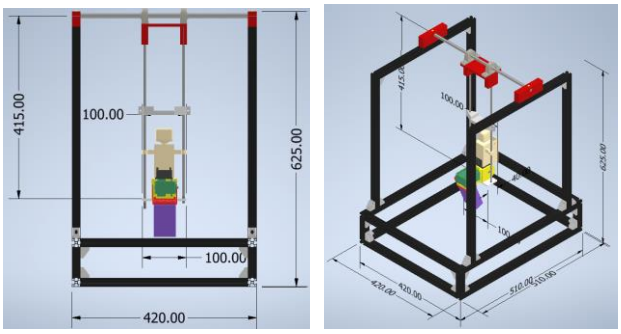

Fig. 5 Design of Swing Robot Frame

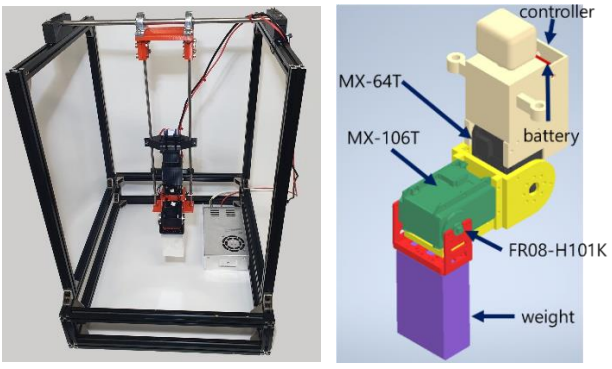

Fig. 6 Design of Swing Robot

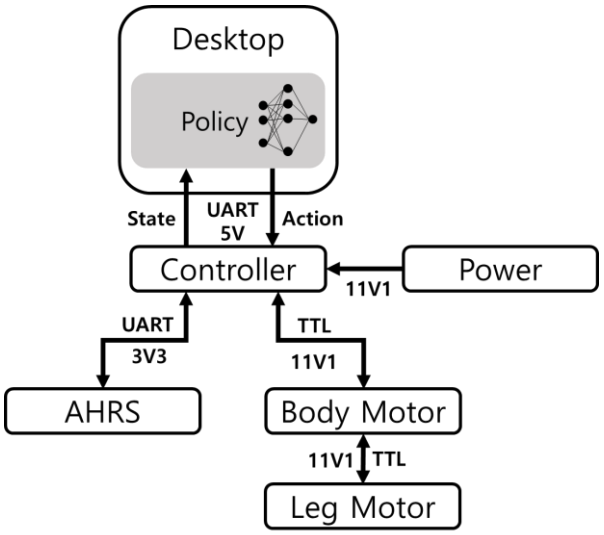

Fig. 7 Diagram of Swing Robot System

5. 시뮬레이션의 구현

실제 기구를 이용한 학습을 하기 전, 알고리즘의 효용성을 검증하기 위하여 시뮬레이션을 구현하고 에이전트를 학습시켰다.

시뮬레이션의 그래픽은 OpenAI 의 Gym 에서 Acrobot-v1 라이브러리를 수정하여 구현하였다.

동역학 모델을 질점으로 가정하고 각 링크에 대하여 라그랑지안 방정식을 세운 뒤, 각 링크의 토크 값을 입력으로 하여 방정식을 Runge-Kutta 4th method 로 해석하여 다음 state 에서 link 의 위치를 예측하였다.

Link1: $T_1 = \frac{1}{2}m_1\left(l_{c1}\dot{\theta}_1\right)^2 + \frac{1}{2}I_1\dot{\theta}_1^2$

$U_1 = m_1 g\, l_{c1}\cos(\theta_1)$

Link2: $T_2 = \frac{1}{2}m_2\left(l_{c1}\dot{\theta_2}\right)^2 + \frac{1}{2}I_2\left(\dot{\theta}_1 + \dot{\theta}_2\right)^2$

$U_2 = m_2 g(l_2\cos(\theta_2) + l_{c2}\cos(\theta_1 + \theta_2))$

$Lagrangian:\ L = T - U = T_1 + T_2 - U_1 - U_2$

$$Lagrangian\ Equation: \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_j}\right) - \frac{\partial L}{\partial \theta_j} = \tau_j (j = 1,2)$$

$$\therefore \begin{pmatrix}\tau_1\\ \tau_2\end{pmatrix} = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix}\begin{pmatrix}\ddot{\theta}_1\\ \ddot{\theta}_2\end{pmatrix} + \begin{bmatrix}C_1\\ C_2\end{bmatrix}\begin{pmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{pmatrix} + \begin{bmatrix}G_1\\ G_2\end{bmatrix}$$

$$M_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + lc_2^2 + 2l_1 l_{c2} C_1) + I_1 + I_2$$
$$M_{12} = M_{21} = m_2(lc_2^2 + l_1 l_{c2} C_2) + I_2$$
$$M_{22} = m_2 lc_2^2 + I_2$$

$$C_1 = -m_2 l_1 l_{c2}\dot{\theta}_2^2 \sin(\theta_2) - 2m_2 l_1 l_{c2}\dot{\theta}_1\dot{\theta}_2 \sin(\theta_2)$$
$$C_2 = m_2 l_1 lc_2\dot{\theta}_1^2 \sin(\theta_2)$$
$$G_2 = m_2 l_{c2} g \cos\left(\theta_1 + \theta_2 - \frac{\pi}{2}\right)$$
$$G_1 = (m_1 l_{c1} + m_2 l_1) g \cos\left(\theta_1 - \frac{\pi}{2}\right) + G_2$$

실제 모터는 단순히 ON/OFF 동작을 하지 않고 모터 프로파일을 이용하여 가감속 운동을 한다. 단순히 Torque 를 ON/OFF 로 구현하게 되면 큰 충격량이 발생하기 때문에 실제 모터와 유사한 거동을 보일 수 있도록 시뮬레이션 상의 모터 프로파일을 작성하였다. 시뮬레이션은 모터가 움직일 방향(CW/CCW)을 입력 받으면 현재 모터의 각도에 따라서 다른 토크를 계산하여 시뮬레이션의 다음 상태를 예측하는 방식으로 이산입력을 받지만 연속출력을 낼 수 있도록 하였다. 모터 프로파일은 Fig. 8 과 같이 각도의 범위를 3 분할하여 구현하였다.

Table 4. Symbol of Dynamics

m_1	$mass\ of\ link1\ [kg]$
m_2	$mass\ of\ link2\ [kg]$
l_1	$length\ of\ link1\ [m]$
l_2	$length\ of\ link2\ [m]$
l_{c1}	$position\ of\ center\ of\ mass\ of\ link1$
l_{c2}	$position\ of\ center\ of\ mass\ of\ link2$
I_1	$moments\ of\ inertia\ of\ link1$
I_2	$moments\ of\ inertia\ of\ link2$
M	$inertia\ matrix$
C	$centrifugal\ and\ coliolis\ matrix$
G	$gravity\ matrix$

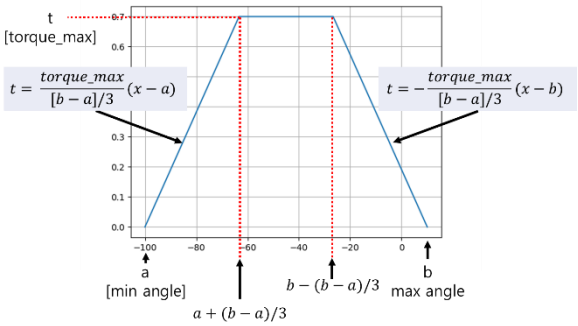

Fig. 8 Motor Profile of Simulation

학습은 한 episode 당 1000 step 을 진행하는데, 특정 종료 조건을 만족할 때까지 반복하여 학습한다.

제어공학적으로 이상적인 모터 제어를 하였을 때 로봇의 스윙 모션은 Fig. 9 와 같은 형태의 토크와 각도 데이터가 출력된다. (3)

따라서 이상적으로 움직이는 그네의 각도 추이는 정현파가 되는 것이 가장 자연스럽다는 것에 착안하여 FFT 를 수행한 결과에서 특정 주파수가 지배적으로 검출되고 20 도 이상 스윙 하는 것이 가능할 때 학습을 종료하도록 설정하였다.

Fig. 9 의 그네 각도와 모터 토크의 추이를 비교하면 모터의 방향 전환 시점과 그네의 각가속도가 변하는 시점이 일치한다. 따라서 일정한 주기로 그네를 제어하게 되는 것을 적절하게 학습하였다는 것의 기준으로 설정하였다.

주파수가 지배적이라는 것의 기준은 가장 큰 크기를 가지는 주파수가 전체 평균 주파수 크기에 비하여 80 배 이상 큰 주파수가 존재할 때이다.

실제로 구현한 동역학 시뮬레이션의 그래픽은 Fig. 10 과 같다.

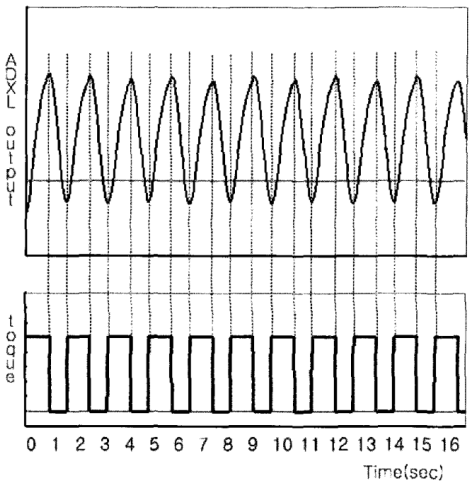

Fig. 9 Ideal Movement Goal (3)

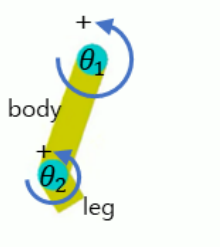

Fig. 10 Model for Simulation

6. 시뮬레이션 및 실제 실험 결과

약 40 번의 시뮬레이션과 실제 기구를 통한 실험을 통하여 적절한 learning rate 가 1.2×10^{-4}임을 찾았다. Learning rate 가 지나치게 큰 경우, reward 가 큰 분산을 가지고 진동하는 모습을 확인할 수 있었다. 반대로 지나치게 작은 경우에는 reward 가 local minima 에 수렴하여 성능 개선 시도 없이 약 6°에 머무르는 것을 확인하였다.

Fig. 11~13 은 한 에피소드에서 진행한 거동을 분석한 그래프이다. 상단은 각도 데이터, 중간은 FFT 데이터, 하단은 모터의 Action 데이터이다.

하단의 Fig 에서 most freq 란 FFT 를 수행한 결과에서 가장 큰 크기를 가지는 주파수를 의미한다. Sigma 란 가장 큰 크기를 가지는 주파수의 크기를 평균 주파수들의 크기로 나눈 값을 의미한다.

$$Sigma = \frac{|most\ freq|}{\frac{1}{n}\sum_{i=1}^{n}|freqency|}$$

다음 Fig.11, Fig.12 는 각각 학습 초기와 학습 후반에서의 결과 데이터이다.

Fig.11 은 시뮬레이션 모델의 학습 초반에 해당하는 2000 episode 의 성능 테스트 데이터이다. 각도 데이터를 보면 일정한 규칙 없이 거동하여 약 4 초경에 최대 각도인 6°까지 올라가고 이후 불규칙적인 동작으로 최대 각도의 개선이 없는 것을 확인할 수 있다.

Fig.12 는 시뮬레이션 모델의 학습 후반에 해당하는 25000 episode 의 테스트 데이터이다. 각도 데이터에서 비교적 균일한 sinusoidal 이 관찰되는 것을 확인할 수 있다. 그네의 방향 전환 시점과 다리의 방향 전화 시점 역시 대부분 일치하는 특정한 규칙이 존재하고 이 규칙을 통하여 일정하게 거동하는 사실을 알 수 있다. 그네가 1000 episode(40s) 동안 최대 20°까지 올라가는 것이 가능한 것을 확인하였고, 목표 성능은 약 4000 episode 에서 만족하였고 학습이 중단되었다. 추가적인 학습을 진행하여 결과 그 이상의 각도 개선은 없는 것으로 확인되었다.

학습을 종료할 때의 Sigma 는 약 80~100 사이로 어느정도 지배적이라고 할 수 있는 주파수를 찾은 것으로 판단하였다.

따라서 시뮬레이션을 이용한 학습으로 실제적으로 그네의 swing motion 을 학습시키는 것이 가능하다는 결론을 내렸다.

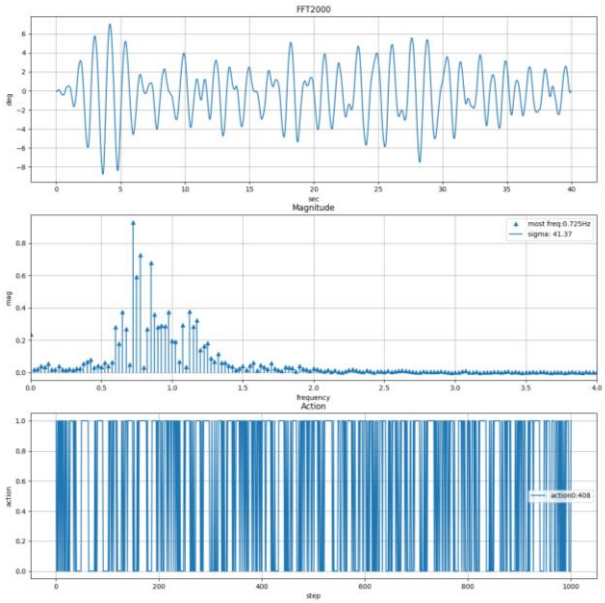

Fig. 11 Training Result of Episode 2000 at Simulation

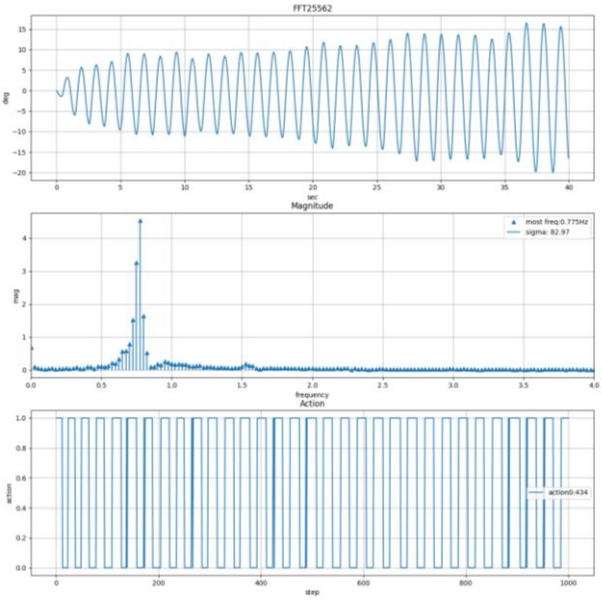

Fig. 12 Training Result of Episode 24000 at Simulation

Fig.13 은 실제 기구로 학습한 데이터이다. 약 4000 episode 까지 학습을 진행하였다. 목표 성능인 20°는 약 2000 episode 전후에서 만족시켰고, 추가적으로 학습을 더 진행한 결과 최대 30°까지 그네가 올라갈 수 있음을 확인하였다. Sigma 는 100 까지 상승하였고 지배적인 주파수는 0.775Hz 가 검출되었다.

Action 의 간격이 시뮬레이션보다 균일하지 않게 보이는 이유는 실제 그네의 무게중심은 정확히 중간에 있지 않기 때문에 그네가 앞으로 움직일 때와 뒤로 움직일 때 무게중심이 많이 흔들리기 때문이다. 따라서 실제로 측정한 결과 앞으로 움직일 때와 뒤로 움직일 때 내려오는 시간이 다른 것을 확인하였다.

시뮬레이션과 실제 기구를 이용한 학습은 Fig. 11 과 같이 유사한 Reward 증가 추이를 보였다. 약 4200 episode 동안 최대 60~100 사이의 reward 까지 상승할 수 있는 것을 확인하였다.

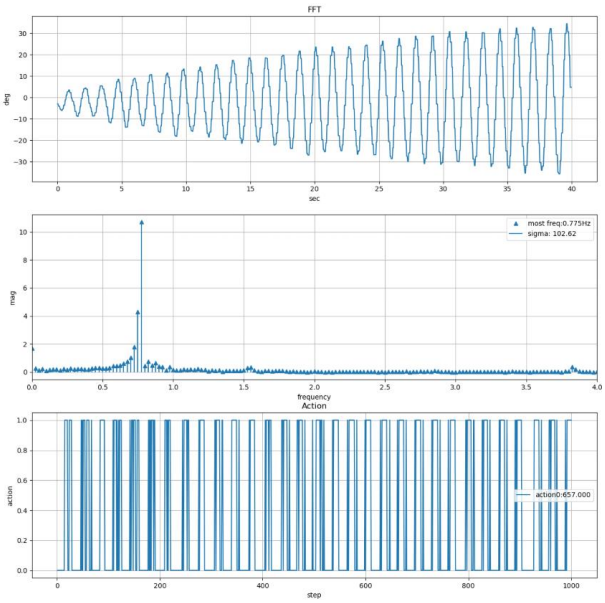

Fig. 13 Training Result of Episode 4000 at Experience

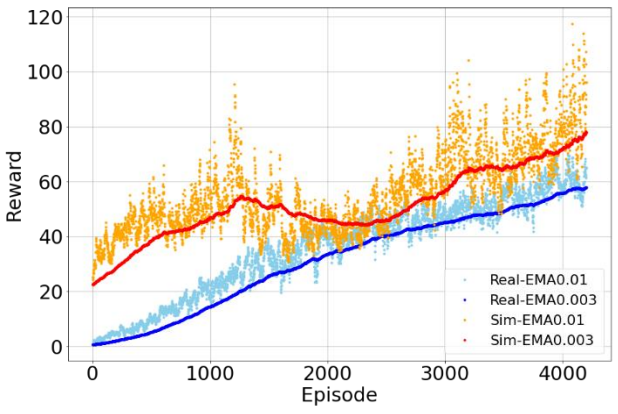

Fig. 14 Rewards of Simulation and Experience

III. 결 론

본 연구의 의의는 고도의 제어공학적 기술을 반복 학습을 통한 기계학습 모델로 대체하는 것에 있다. 이를 검증하기 위해 시뮬레이션과 실제 기구를 제작하고 각각 Actor-Critic 알고리즘을 이용한 강화학습으로 Swing motion 을 만들어 낼 수 있음을 증명하였다. 향후에는 서서 타는 그네와 상체 모터를 이용한 다자유도 로봇 제어 연구를 진행할 계획이다.

참고문헌

[1] H.S. Choi, D.W. Kim, H.H. Park, Simulation of swing robot using reinforcement learning, KSMTE, 192 1-2, 2021

[2] Shalabh Bhatnagar, Richard Sutton, Mohammad Ghavamzadeh, Mark Lee, Natural Actor-Critic Algorithms, Automatica, 2471-2482, 2009

[3] S.H. Park, S.Y. Yi, K.T. Chong, Y.W. Sung, 2004, Swing Motion of Miniaturized Humanoid robot, Journal of Institute of Control, Robotics and Systems, 10(3), March 2004, 267-272