공 학 석 사 학위논문

강화학습을 이용한 오프셋 데이터 최적 보간

Offset Point Interpolation using Reinforcement Learning

울 산 대 학 교 대 학 원
조 선 및 해 양 공 학 과
장 연 진

강화학습을 이용한 오프셋 데이터 최적 보간

지 도 교 수 오 민 재

이 논문을 공학석사학위 논문으로 제출함

2024 년 02 월

울 산 대 학 교 대 학 원
조 선 및 해 양 공 학 과
장 연 진

장연진의 공학석사학위 논문을 인준함

심사위원 신 동 목 

심사위원 오 민 재 

심사위원 김 기 수 

울 산 대 학 교 대 학 원

2024 년 02 월

강화학습을 이용한 오프셋 데이터 최적 보간

장 연진
울산대학교 일반대학원 조선및해양공학과

국문 요약

B-spline 곡선 보간(B-spline curve interpolation)은 컴퓨터 그래픽스, 컴퓨터 지원 설계 및 로봇공학과 같은 다양한 분야에서 널리 사용된다. 보간된 곡선의 형상은 매개변수화 방법(Parameterization method)에 따라 계산한 제어점(Control point)에 크게 의존한다. B-스플라인 곡선 보간 중 보간된 곡선의 형상에 영향을 주는 매개변수 값(Knot)을 구할 때 사용되는 매개변수화 방법에 따라 제어점이 다르게 계산되며, 이러한 특징으로 인해 원하는 곡선의 형태를 정확하게 나타내기 어렵다. 이러한 한계를 극복하기 위해 본 연구에서는 B-스플라인 곡선 보간을 수행할 때 강화학습을 기반으로 한 새로운 매개변수 최적화 방법을 제안한다. 제안된 방법을 선형을 나타내는 오프셋 데이터에 적용하여 계산한 매개변수 값을 기반으로 단면 곡선(Section curve)을 생성한다. 강화학습의 에이전트(Agent)는 0에서 1까지의 범위로 정규화된 매개변수를 반복적으로 조절하고 변환된 매개변수 값을 상태로 정의한다. 에이전트는 상태 변화를 관찰하고 환경(Environment)은 결과 곡선을 관찰하여 보상을 주는 방법을 통하여 주어진 점에 대한 최적의 매개변수 값을 학습하게 된다. 매개변수 값을 반복적으로 조절하고 결과 곡선을 관찰함으로써 학습 에이전트는 주어진 B-스플라인에 대한 최적의 매개변수 값을 학습한다. 제안된 방법을 평가하기 위해 기존의 최적화 방법인 유전 알고리즘 모델을 사용하여 얻은 결과를 제안된 알고리즘을 적용한 결과와 비교하였다. 또한, 오프셋 데이터를 기반으로 단면 곡선을 생성할 때 기존 매개변수화 방법과 제안된 방법을 비교하고, 학습된 모델을 다른 선종의 오프셋 데이터에 적용하여 제안된 방법의 유용성을 확인하였다. 제안하는 매개변수화 최적화 방법이 기존 매개변수화 방법보다 더 우수한 성능을 나타내는 것을 확인하였다. 본 연구에서 제안하는 방법은 B-spline 곡선을 보간할 때 최적의 매개변수를 찾기 위한 새로운 접근 방식으로, 기하 모델링에서 머신러닝 기법을 활용할 수 있는 잠재력을 보여준다.

목차

그림 목차

표 목차

제1장 서론

1.1. 연구 배경

4차 산업혁명 시대에 들어서며, 머신러닝 알고리즘을 활용한 연구가 다양한 산업 분야에서 진행되고 있다. 머신러닝이란 인공지능의 하위 분야로, 컴퓨터 프로그램이 데이터로부터 학습하고, 이를 통해 패턴을 파악하고 결정을 내리는 기술이다. 이러한 기술 중 강화학습이란, 다른 머신러닝 패러다임과는 대조적으로 특정한 데이터 집합에 의존하지 않고 인공 신경망이 주어진 환경에서 직접적인 탐색과 시행착오를 통해, 각 상태에서 얻을 수 있는 보상을 최대화하기 위한 행동을 학습하는 방법론이다.[1] 본 연구에서는 순차적 의사결정 문제인 오프셋 데이터 최적 보간 과정을 마르코프 결정 프로세스의 수학적 모델로 형상화하고, 이를 강화학습 알고리즘을 적용하여 오프셋 데이터 최적 보간 문제를 해결하는 방법을 제안한다.

선박의 선형을 나타내는 오프셋 데이터는 Fig. 1.1.과 같이 보간 과정을 거쳐 section curve로 가시화 할 수 있으며 section curve를 이용하여 body plan을 그릴 수 있다.

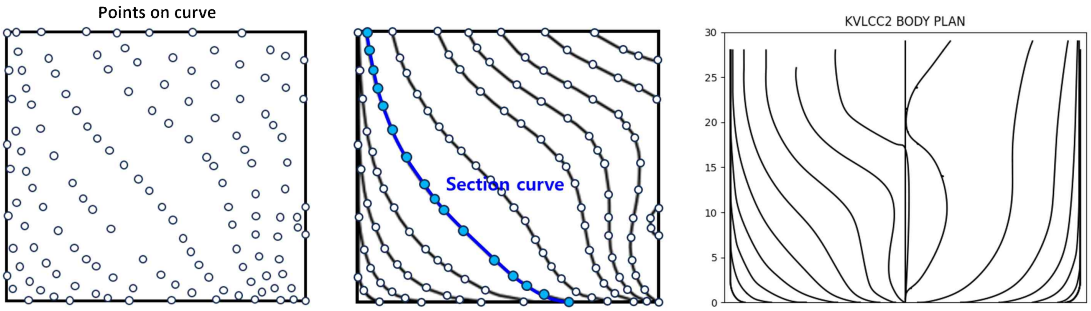

Fig. 1.1. 오프셋 데이터를 이용하여 section curve와 body plan을 구현한 예

주어진 오프셋 데이터를 지나는 선형을 생성하기 위해 섹션 별로 B-spline 곡선 보간(interpolation)을 수행한다[2][3]. Fig. 1.2.는 본 연구에서 이용한 B-spline 곡선 보간 과정을 나타낸 것이다. 주어진 점 데이터를 기반으로 매개변수화 방법 중 가장 정확하다고 알려진 centripetal 매개변수화 방법[4]으로 매듭(knot)값을 계산하다. 이후, 처음과 마지막 매듭 값에 차수(degree)인 3만큼 매듭을 삽입(knot insertion)[2]한다. 계산한 매듭 값을 이용하여 Bessel end condition[2]을 이용하여 첫 번째와 n-1번째 제어점(control point)를 계산한다. 매듭 값과 차수를 이용하여 B-spline 기저 함수를 계산하여 기저 행렬을 정의한 후, 기저 행렬과 주어진 점 데이터를 이용하여 B-spline 곡선 보간을 수행한다.

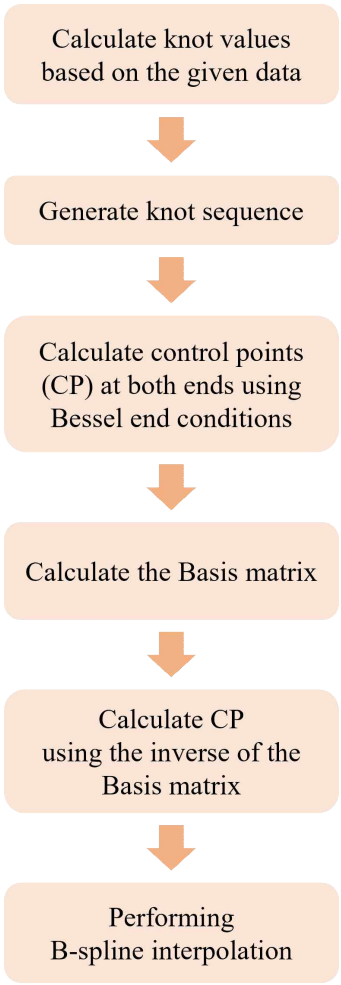

Fig. 1.2. Flow chart of the B-spline interpolation

B-spline 곡선 보간에서 매듭 값을 계산하는 방법인 매개변수화 방법마다 계산되는 매듭 값이 달라진다. Fig. 1.3.은 매개변수화 방법에 따라 다르게 계산된 매듭 값을 나타낸다.

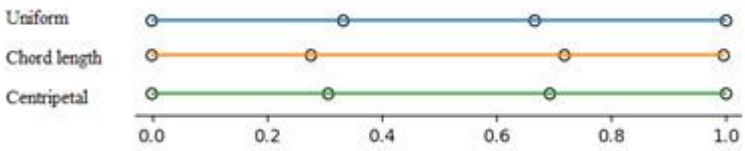

Fig. 1.3. 기존의 매개변수화 방법을 이용한 매듭 값

매듭 값이 다르기 때문에 제어점과 기저함수가 다르게 계산되므로, Fig. 1.4.와 같이 보간된 곡선의 형상 또한 이용한 매개변수화 방법에 따라 달라진다. 이와 같이 원하는 형상을 표현하기 위한 최적의 매개변수를 찾기 어려운 경우가 있다. 이러한 제약을 극복하기 위해, 본 연구에서는 강화학습을 오프셋 데이터 최적 보간 문제에 적용하였다. Fig. 1.5.와 같이 인공지능이 매듭 값을 변환한 후 변환된 제어점과 기

저함수를 이용해 모델을 학습하여 기존의 매개변수화 방법을 이용하였을 때보다 정교한 B-spline 곡선 보간이 가능한 객관적인 최적화 모델을 만들 수 있을 것이라 기대한다.

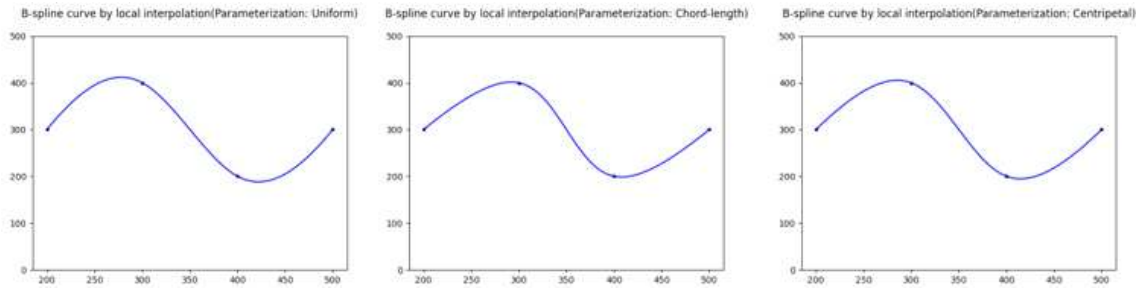

Fig. 1.4. 매개변수화 방법을 이용하여 보간한 B-spline 곡선

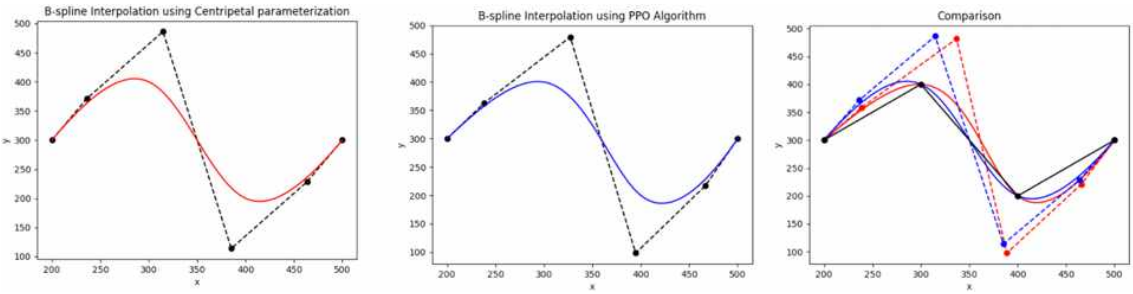

Fig. 1.5. 매개변수화 방법을 이용하여 보간한 B-spline 곡선

해당 모델에 선형을 나타내는 오프셋 데이터를 적용하여 최적의 매듭 값을 구해, Fig. 1.6과 같이 단면 곡선을 생성하고자 한다.

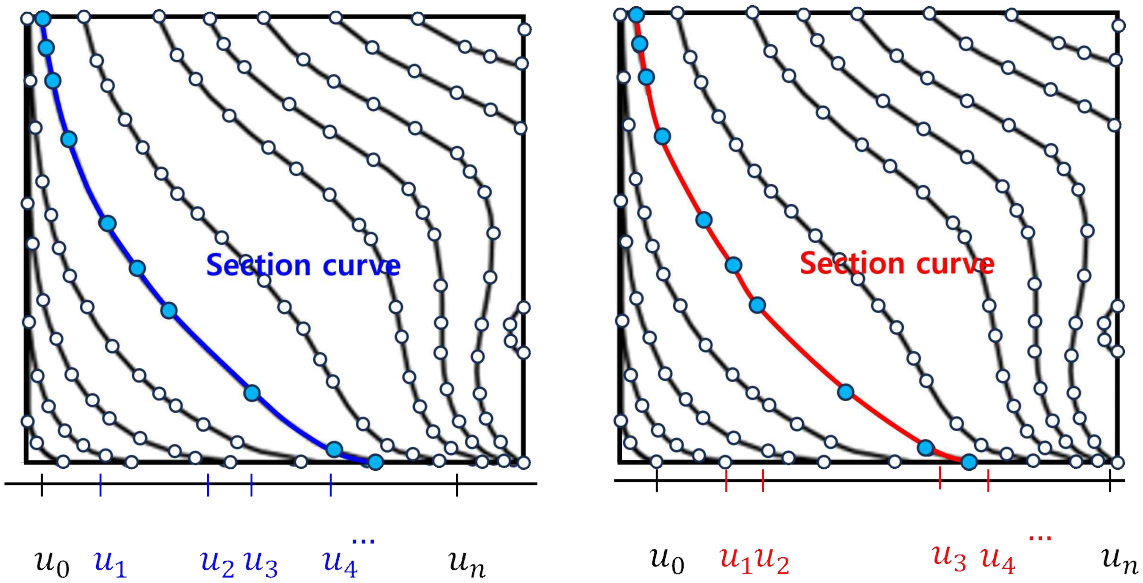

Fig. 1.6. 매듭 값에 따라 다르게 보간된 단면 곡선의 예

1.2. 관련 연구

B-spline 곡선 보간 과정에서 매듭 값을 계산하는 매개변수화 방법에 따라 보간된 곡선의 형상이 달라지는데, 가장 보편적인 매개변수화 방법은 uniform[3], chord length[4], centripetal[5] 매개변수화 방법이다.

$$\frac{\bar{u}_{i+1}-\bar{u}_i}{\bar{u}_i-\bar{u}_{u-1}}=\frac{\triangle_{i+1}}{\triangle_i},\ \triangle_i=\|L_i\|^e, 1\le i\le n \qquad (1.1)$$

여기서, $\bar{u}_0=0$이며 $\|L_i\|$는 주어진 점 데이터 $\mathbf{p}_{i-1}$에서 $\mathbf{p}_i$까지의 노름 벡터(norm vector) L_i이다. uniform 방법은 $e=0$, chord length 방법은 $e=1$, centripetal 방법은 $e=1/2$이다.

이 세 가지 방법은 식(1.1)을 통해 구할 수 있다. Uniform 매개변수화 방법은 곡선을 균일하게 나누기 위해 곡선의 길이를 모든 매듭 값 간격에 균등하게 할당한다. 매듭 값의 간격은 모두 동일하므로 곡선의 주어진 점 데이터 간의 거리나 각도에 따라 변하지 않는다. 이 방법은 간단하지만 곡선의 굴곡과 곡률을 전혀 고려하지 않기 때문에, 원하는 곡선의 형태를 구현하지 못한다는 단점이 존재한다. Chord length 매개변수화 방법은 주어진 점 데이터를 직선으로 이은 거리에 비례하여 매듭 값을 할당한다. 점 데이터가 직선상 거리가 큰 부분에서는 매듭 값이 더 크게 할당된다. 이로 인해 곡선은 점 데이터 사이의 거리가 큰 부분에서 길게 계산되며 적은 부분에서는 짧게 계산된다. 매듭 값이 직선 거리에 따라 계산되므로, 곡선이 복잡하거나 인접한 점 데이터간의 길이가 긴 경우 파라미터 값의 불균형을 초래한다. Centripetal 매개변수화 방법은 점 데이터 간 거리의 제곱근을 사용하여 매듭 값을 계산하는데, 거리가 멀수록 매듭 값이 빠르게 증가하는 경향이 있다. 곡선이 복잡하거나 뾰족한 부분이 있는 경우 안정적인 매듭 값을 얻을 수 있지만 곡선의 길이와 곡률에 따라 매듭 값의 분포가 조정되므로 긴 길이의 곡선이나 굴곡이 많은 부분에서 부적절하다.

Fig. 1.7.과 같이 주어진 점 데이터가 같더라도 어떤 매개변수화 방법을 이용했는지에 따라 보간된 결과는 다르게 나타난다. 또한 주어진 점 데이터의 분포 특징에 따라 적합한 매개변수화 방법이 다르며, 해당 방법을 이용하더라도 최적의 매듭 값이 계산된다는 것은 보장되지 않는다.

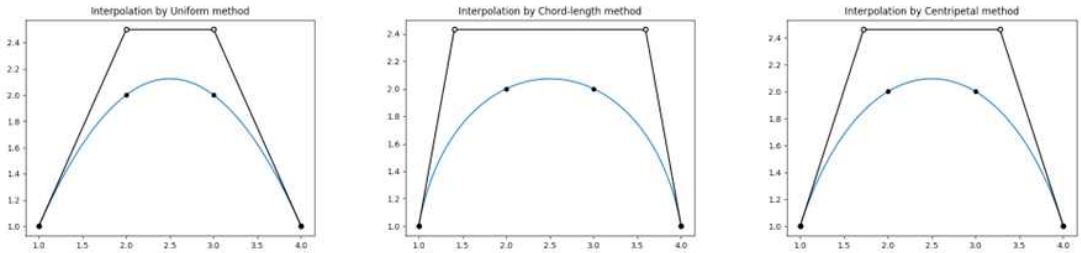

Fig. 1.7. 다른 매개변수화 방법을 이용하여 보간한 3차 Bézier 곡선

이러한 문제점을 해결하기 매개변수화 방법에 최적화 알고리즘이 적용되어야 한다. Hu et al.(2020)은 매개변수화 방법에 유전 알고리즘을 적용한 매개변수화 방법을 제안하였다.[18] 본 연구에서는 매개변수화 계산 과정에 강화학습 알고리즘을 적용하여 오프셋 데이터 최적 보간 방법에 대해 연구한다.

1.3. 연구 목적

B-spline 곡선 보간에서 주요 파라미터인 매듭 값은 이용하는 매개변수화 방법에 따라 다르게 계산되며 매듭 값에 따라 보간된 곡선의 형상 또한 다르게 구현된다. 이로 인해 주어진 점 데이터의 분포에 따라 적합한 매개변수화 방법이 각각 다르다는 문제가 있다.

이러한 문제를 해결하고자, 본 연구에서는 B-spline 곡선 보간에서 매듭 값을 강화학습 알고리즘을 이용하여 수행하고 변환된 매듭 값을 이용하여 B-spline 곡선을 보간하고자 한다. 보간된 곡선과 주어진 점 데이터를 직선으로 이은 data polyline이 이루는 면적 및 곡선의 에너지 값의 변화를 인공지능이 학습하여 정교하고 객관적인 오프셋 데이터 최적 보간 모델을 구현하는 것을 목표로 한다.

강화학습을 이용한 오프셋 데이터 최적 보간 과정은 매듭 값의 변환량을 의사 결정자인 에이전트의 행동(Action)으로 정의하고 변환한 매듭 값을 이용하여 B-spline 곡선 보간을 수행한다. 에이전트는 매듭 값을 관측하고 변화에 대한 보상(Reward)을 통해 다음 행동을 결정한다. 이러한 과정을 반복하여 행동의 선택이 종료되었을 시점까지 누적된 총 보상이 최대가 되면 오프셋 데이터 최적 보간이 완료되었다고 판단한다. 본 연구에서는 강화학습 방법론 중 정책 그래디언트(Policy gradient) 방법을 이용하였다.

가치함수를 함께 추정하여 정책의 성과를 평가해 최적 정책을 구하는 액터-크리틱(Actor-critic) 방식의 알고리즘을 이용해 구현 결과와 기존의 최적화 알고리즘 중 유전 알고리즘(Genetic algorithm)을 이용하여 구현한 결과를 비교 분석하여 성능을 평가한다.

이후 강화학습 알고리즘을 이용하여 학습한 모델을 다른 선종에 적용하여 해당 선박의 오프셋 데이터 최적 보간을 수행한다.

1.4. 논문의 구성

본 논문은 다음과 같이 구성되어 있다. 2장에서는 B-spline 보간에 사용된 이론과 최적화 알고리즘 등의 배경 이론과 간단한 예제에 대하여 설명한다. 3장에서는 본 논문에서 제안하는 심층 강화학습을 이용한 오프셋 데이터 최적 보간에 관한 내용 및 결과에 대해 설명한다. 또한 기존의 최적화 알고리즘으로 사용되는 유전 알고리즘과 심층 강화학습 모델의 결과 비교를 통해 본 연구에서 제안하는 방법의 성능을 검증한다. 이후 강화학습을 이용하여 학습한 모델에 선박의 오프셋 데이터를 적용하여 단면 곡선 및 body plan을 생성한다. 마지막으로 4장에서는 연구 수행을 통한 결론 및 고찰을 정리하였다.

제2장 배경 이론

2.1. Bézier 곡선

Bézier 곡선(Bézier curve)은 컴퓨터 그래픽스와 컴퓨터 이용 설계(CAD) 및 로보틱스와 같은 다양한 분야에서 사용되는 곡선 모델링 기술 중 하나로, 제어점(control point)을 사용하여 부드러운 곡선을 생성하고 데이터를 보간하는 데 활용된다.[2] Bézier 곡선을 정의하는 데 사용되는 제어점은 2D 또는 3D 공간 상의 점이다. 곡선의 차수는 2차, 3차, 또는 더 높은 차수를 가질 수 있다. 이 차수는 곡선의 복잡도를 결정하는데, 차수가 높을수록 더 복잡한 곡선을 형성한다. 가장 일반적인 형태의 Bézier 곡선은 4개의 제어점을 사용하여 곡선을 정의하는 3차 Bézier 곡선이다. 이렇게 정의된 Bézier 곡선은 제어점을 통해 곡선을 형성하며, 두 중간 제어점이 곡선의 모양을 조절한다.

2.1.1. 3차 Bézier 곡선 보간

Bézier 곡선 보간은 주어진 점 데이터를 지나는 곡선을 형성하여 부드럽게 보간을 수행하는 수학적 기술이다. 3차 Bézier 곡선 보간 방법은 다음과 같다. 우선 주어진 점 데이터 $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$를 이용하여 파라미터 t_i를 계산한다. 이때 파라미터는 0에서 1사이의 값으로 정규화되며, 곡선의 형태를 조절하는 데 사용된다. 일반적으로 $\mathbf{p}_1$과 $\mathbf{p}_2$는 t값이 0.0인 시작점에서부터 1.0인 종료점까지의 위치에서 선택되므로 $\mathbf{p}_0$과 $\mathbf{p}_3$의 t값은 각각 0.0과 1.0이 된다.

$$\begin{aligned} B_0^3(t) &= (1-t)^3 \\ B_1^3(t) &= 3t(1-t)^2 \\ B_2^3(t) &= 3t^2(1-t) \\ B_3^3(t) &= t^3 \end{aligned} \tag{2.1}$$

3차 Bézier 곡선의 기저 함수는 t값을 사용하여 제어점($\mathbf{b}_i$)을 계산하는 데 사용된다. 3차 Bézier 곡선의 기저 함수는 식(2.1)과 같이 계산한다.

$$\mathbf{r}(t) = B_0^3(t)\mathbf{b}_0 + B_1^3(t)\mathbf{b}_1 + B_2^3(t)\mathbf{b}_2 + B_3^3(t)\mathbf{b}_3 \tag{2.2}$$

3차 Bézier 곡선을 수식으로 나타내면 식(2.2)와 같다.

$$
\begin{aligned}
\mathbf{p}_0 &= B_0^3(t_0)\mathbf{b}_0 + B_1^3(t_0)\mathbf{b}_1 + B_2^3(t_0)\mathbf{b}_2 + B_3^3(t_0)\mathbf{b}_3 \\
\mathbf{p}_1 &= B_0^3(t_1)\mathbf{b}_0 + B_1^3(t_1)\mathbf{b}_1 + B_2^3(t_1)\mathbf{b}_2 + B_3^3(t_1)\mathbf{b}_3 \\
\mathbf{p}_2 &= B_0^3(t_2)\mathbf{b}_0 + B_1^3(t_2)\mathbf{b}_1 + B_2^3(t_2)\mathbf{b}_2 + B_3^3(t_2)\mathbf{b}_3 \\
\mathbf{p}_3 &= B_0^3(t_3)\mathbf{b}_0 + B_1^3(t_3)\mathbf{b}_1 + B_2^3(t_3)\mathbf{b}_2 + B_3^3(t_3)\mathbf{b}_3
\end{aligned}
\tag{2.3}
$$

식(2.2)를 이용하여 보간 조건(interpolation conditions)을 나타내면 식(2.3)과 같이 4개의 미지수 $\mathbf{b}_i (i=0,1,2,3)$과 4개의 식을 얻을 수 있다.

$$
\begin{bmatrix} \mathbf{p_0} \\ \mathbf{p_1} \\ \mathbf{p_2} \\ \mathbf{p_3} \end{bmatrix} = \begin{bmatrix} B_0^3(t_0)\, B_1^3(t_0)\, B_2^3(t_0)\, B_0^3(t_3) \\ B_0^3(t_1)\, B_1^3(t_1)\, B_2^3(t_1)\, B_0^3(t_3) \\ B_0^3(t_2)\, B_1^3(t_2)\, B_2^3(t_2)\, B_0^3(t_3) \\ B_0^3(t_3)\, B_1^3(t_3)\, B_2^3(t_3)\, B_0^3(t_3) \end{bmatrix} \begin{bmatrix} \mathbf{b_0} \\ \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \end{bmatrix}
\tag{2.4}
$$

미지수인 제어점 $\mathbf{b}_i$를 계산하기 위해 3차 Bézier 곡선식을 행렬로 나타내면 식(2.4)와 같고, 식(2.4)는 $\mathbf{P}=\mathrm{M}\mathbf{B}$의 형태로 나타낼 수 있다. 따라서 구해야 하는 제어점은 양변에 M^{-1}을 곱하여$\mathbf{B}=\mathrm{M}^{-1}\mathbf{P}$의 식을 통해 구할 수 있다.[2]

2.2. B-spline 곡선

B-spline 곡선은 2.2.1절에서 서술 할 C^2 연속성을 만족하는 3차 Bézier 곡선들로 이루어져있으며, 다항식(polynomial) 함수의 조합으로 정의된다.[2] 이러한 다항식 함수들을 기반으로 B-spline 곡선은 부드러운 곡선을 생성할 수 있다. B-spline 곡선의 제어점의 위치를 조절함으로써 곡선의 형태를 다양하게 조절할 수 있다. B-spline 곡선 또한 차수(degree)를 가지는데, 차수는 곡선의 부드러움 및 복잡성을 조절한다. 일반적으로 낮은 차수의 B-spline 곡선은 직선에 가깝고, 곡선의 차수가 증가함에 따라 복잡한 곡선을 나타낸다.

B-spline 곡선은 곡선을 이루는 다중 세그먼트를 통해 복잡한 경로 및 형태를 만들 수 있다. 또한, 제어점의 위치 및 가중치를 조절하여 곡선의 매끄러움을 조절할 수 있는데, 이는 곡선이 제어점을 통과하면서도 부드럽게 이어지도록 한다. B-spline 곡선은 다양한 분야에서 곡선의 부드러움과 유연성을 통해 복잡한 형태와 경로를 모델링하는 데 효과적으로 활용된다.

2.2.1. 곡선의 연속성

C^{-1} 연속성은 Fig. 2.1.과 같이 두 개의 곡선이 이어져 있지 않은 것을 의미한다. $f_1(t) = a_0 + a_1 t + a_2 t^2$, $f_2(t) = b_0 + b_1 t + b_2 t^2$라 정의할 때, $f_1(1) \neq f_2(1)$, 이다.

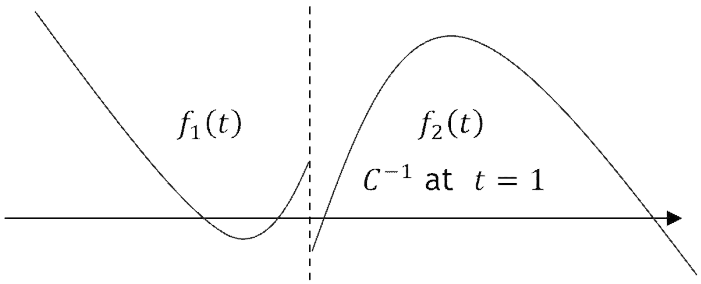

Fig. 2.1. C^{-1} 연속성

C^0 연속성은 Fig. 2.2.와 같이 두 개의 연속된 곡선이 서로 부드럽게 이어져 보이는 것을 의미한다. $f_1(1) = f_2(1)$로, 두 개의 곡선이 접합 지점에서 같은 지점을 공유하고 연속된 형태를 이루어야 한다.

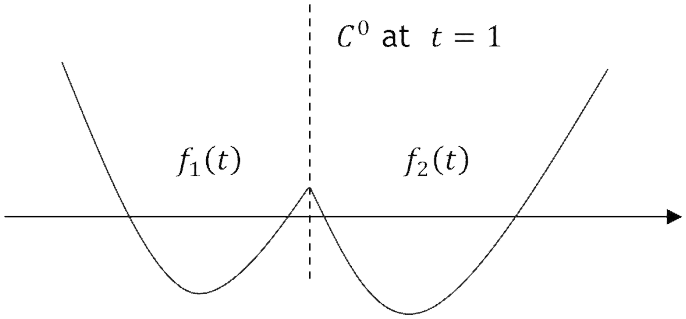

Fig. 2.2. C^0 연속성

C^1 연속성은 Fig. 2.3.과 같이 C^0 연속성을 포함하면서, 접합 지점에서의 접선이 일치한다는 것을 나타낸다. $f_1(1) = f_2(1)$, $f_1'(1) = f_2'(1)$로, 두 곡선이 접합 지점에서 이어지는 부드러운 곡선을 형성한다. 이 지점에서 두 곡선의 1차 미분 값, 즉 접선 방향이 동일하다.

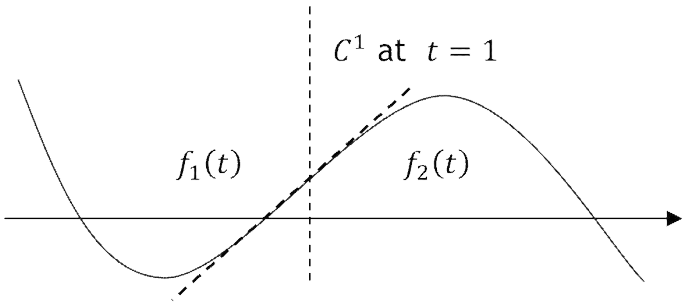

Fig. 2.3. C^1 연속성

C^2 연속성은 Fig. 2.4.와 같이 C^1 연속성을 포함하면서, 접합 지점에서의 곡률이 연속된 것을 나타낸다. $f_1(1) = f_2(1)$, $f_1'(1) = f_2'(1)$, $f_1''(1) = f_2''(1)$로, 두 곡선이 접합 지점에서 이어지는 부드러운 곡선을 형성한다. 이 지점에서의 2차 미분 값, 즉 곡률이 일관되게 변하지 않는다.

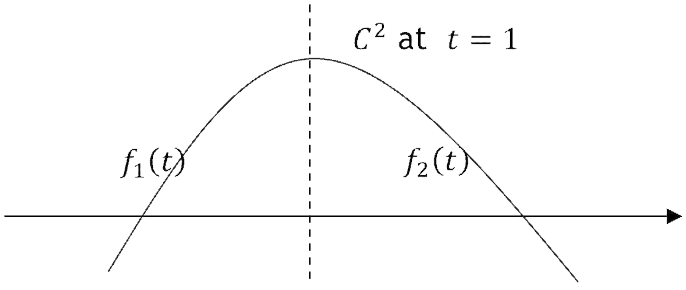

Fig. 2.4. C^2 연속성

이러한 연속성은 곡선이 서로 연결될 때 부드러운 정도를 정의하며 연속성의 수준이 높을수록, 연결된 곡선이 더 부드럽고 자연스러운 모양을 가진다고 볼 수 있다.

2.2.2. B-spline 곡선 보간

보간(interpolation)기법은 주어진 점 데이터들 사이의 부드러운 곡선을 생성하는 과정이다.

Fig. 2.5.와 같이 점 데이터가 주어진 경우, B-spline 곡선 보간을 수행하여 데이터 포인트 사이의 값을 계산하고 해당 값을 이용하여 Fig. 2.6.과 같이 부드럽게 연결된 곡선을 얻을 수 있다.

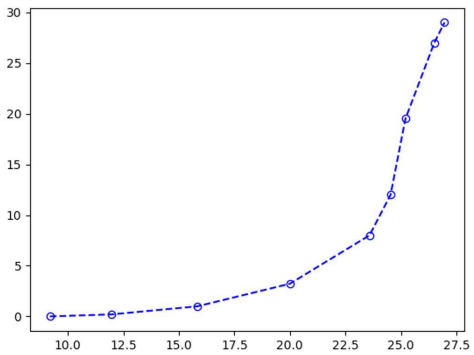

Fig. 2.5. Data polyline

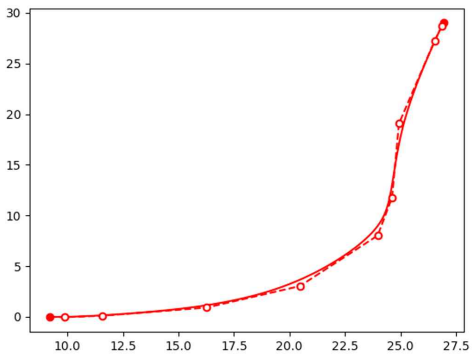

Fig. 2.6. B-spline curve

B-spline 곡선 보간의 순서는 Fig. 2.7.과 같다. 주어진 점 데이터를 기반으로 매듭(knot)값을 계산한다. 매듭 값은 B-spline 곡선을 정의하는 중요한 파라미터 값으로, 점 데이터와 제어점 사이의 간격을 결정한다. 보간 할 차수에 맞게 매듭 값을 이용하여 매듭 시퀀스(knot sequence)를 생성한다. 이후 Bessel end condition을 이용하여 양 끝의 제어점을 계산하고 B-spline 기저 함수(basis function)를 정의한다. B-spline 기저 함수로 구성된 행렬 M의 역행렬 M^{-1}와 주어진 점 데이터를 이용하여 만든 행렬 P를 곱하여 나머지 제어점을 계산한 후, B-spline 곡선 보간을 수행한다.

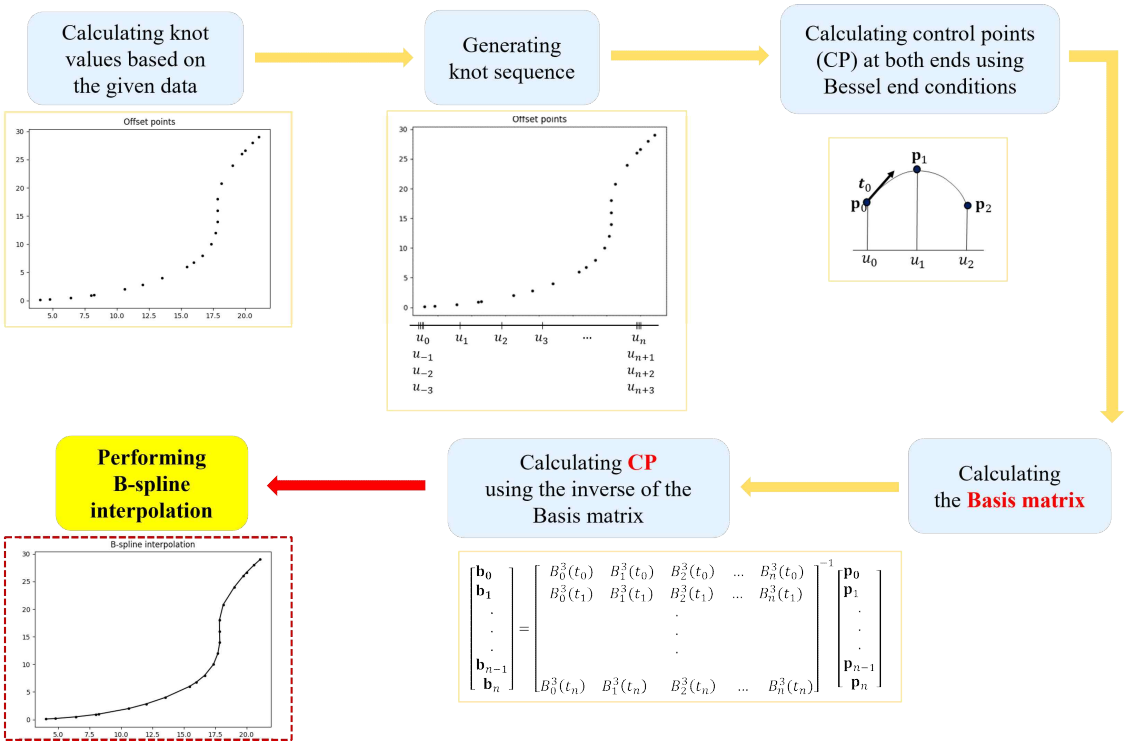

Fig. 2.7. B-spline 곡선 보간 과정

2.2.3. B-spline 기저 함수

본 연구에서는 Cox-de Boor 재귀 공식을 이용하여 B-spline 기저 함수를 정의하였다.[2]

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u)$$
$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

여기서, $U = \{u_0, \ldots, u_m\}$, $u_i \le u_{i+1}$, $i = 0, \ldots, m-1$이고 이때, u_i는 매듭 값이며 U는 매듭 벡터(knot vector)이다.

식 (2.5)는 Cox-de Boor 재귀 공식이다. 차수를 p로 정의하면, i번째 B-spline 기저 함수, $N_{i,p}(u)$는 식(2.5)와 같이 정의된다. $p>0$인 경우, $N_{i,p}(u)$는 두 개의 $p-1$차 함수의 선형 결합이다.

2.2.4. Knot Insertion

Bézier 곡선으로 이루어진 B-spline 곡선은 매듭 삽입(Knot Insertion)을 사용하여 기존의 B-spline 곡선의 모양을 바꾸지 않고 여러 개의 Bézier 곡선으로 나눌 수 있다. 새로운 매듭 값을 삽입하면 새로운 제어점이 추가된다.[2]

본 연구에서는 Fig. 2.8.과 같이 매듭 삽입을 통해 B-spline 곡선을 여러 개의 Bézier 곡선으로 나눈 후 회전변환과 적분을 이용하여 본 연구에서 제안하는 모델을 구현할 때 설정한 강화학습의 보상 값 중 하나인 data polyline과 B-spline 곡선 간의 면적을 계산하였다.

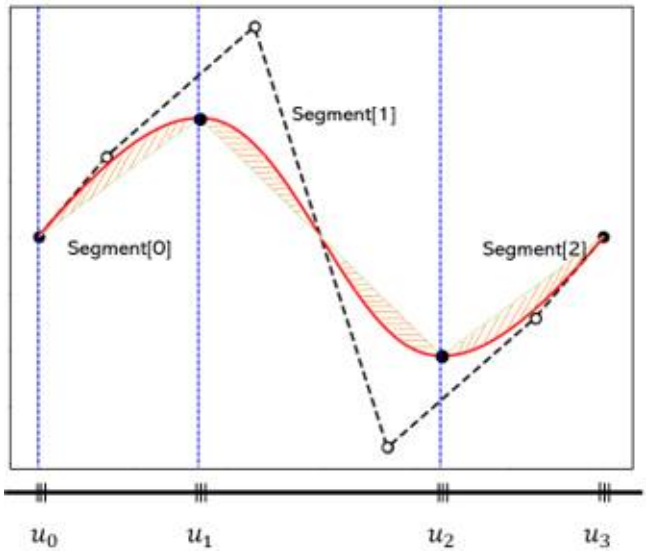

Fig. 2.8. Knot 삽입을 이용하여 B-spline 곡선을 Bézier 곡선으로 나눈 예

2.2.5. Bessel End Condition

본 연구에서는 보간 조건(Interpolation Condition)으로 계산할 수 없는 두 제어점 $\mathbf{d}_1$과 $\mathbf{d}_{n-1}$을 Bessel End Condition을 이용하여 구하였다.[6]

Bessel End Condition은 Fig. 2.9.과 같이 나타낼 수 있다. B-spline 곡선 보간에서 양 끝점의 접선벡터 t_0과 t_1이 주어지지 않은 경우 곡선의 양끝의 연속된 세 점을 이용하여 2차 곡선(quadratic curve)을 생성한 후, 생성된 2차 곡선의 양 끝점에서의 1차 미분 값을 B-spline 곡선의 양 끝점에서의 접선 벡터로 가정하는 기법이다.

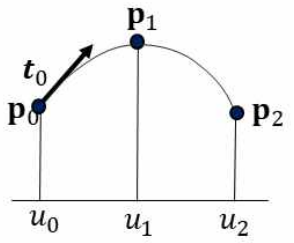

Fig. 2.9. Bessel End Condition

$\mathbf{p}_0$에서의 일차 미분 값을 이용하여, $\mathbf{d}_1$은 식(2.6)을 통해 구할 수 있으며 반대쪽 제어점인 $\mathbf{d}_{n-1}$도 같은 방법으로 구할 수 있다.

$$
\begin{aligned}
&\triangle_i = u_{i+1} - u_i \\
&\alpha = \frac{\triangle_1}{u_2 - u_0}, \beta = 1 - \alpha \\
&\mathbf{a} = \frac{1}{2\alpha\beta}(\mathbf{p}_1 - \alpha^2\mathbf{p}_0 - \beta^2\mathbf{p}_2) \\
&\mathbf{d}_1 = \frac{2}{3}(\alpha\mathbf{p}_0 + \beta\mathbf{a}) + \frac{1}{3}\mathbf{p}_0
\end{aligned}
\tag{2.6}
$$

2.3. Curve Energy

곡선이 부드럽게 보간 되었는지 평가하는 방법 중 하나는 보간된 곡선의 에너지를 평가하는 것이다. 에너지가 작을수록 특정 지점에서 휘어지는 구간이 없이 보간 되었다고 볼 수 있다.

$$E_{bend}(t) = \int \kappa^2 ds \qquad (2.7)$$

곡선의 에너지는 식 (2.7)을 통해 구할 수 있다.[7][8] 이때, κ는 곡선의 곡률이고 s는 호의 길이이다. 보간된 곡선은 불필요한 진동이나 왜곡된 부분이 없어야하며, 곡선이 부드러운 형상으로 보간 되었는가에 관한 문제는 중요하다. 따라서 본 연구에서는 보간된 곡선의 부드러움을 평가하기 위해, 보간된 곡선의 에너지 값을 계산하여 보상 값으로 설정하였다.

2.4. 최적화 알고리즘

최적화 알고리즘은 다양한 문제에서 최상의 솔루션을 탐색하는 데 사용되는 방법론이다. 최적화 알고리즘은 주어진 조건 하에서 목적 함수를 최대화 또는 최소화하는 최적의 매개변수 집합을 탐색하는 데 사용된다. 이때, 목적 함수는 특정 문제의 성능을 나타내며, 최적화 알고리즘은 목적함수를 최적화하는 매개변수 조합을 결정하여 최상의 솔루션을 찾아내는데 활용된다. 최적화 알고리즘은 적용할 문제의 복잡성과 특성에 따라 적합한 방법을 선택하여 사용된다. 본 연구에서는 정책 그래디언트를 이용하여 오프셋 데이터 최적 보간 모델을 구현하였으며, 기존의 최적화 알고리즘인 최적화 알고리즘을 예제 데이터에 적용하였다.

제3장 강화학습을 이용한 오프셋 데이터 최적 보간

3.1. 심층 강화학습 알고리즘(DRL)

강화학습에서 의사결정자인 에이전트가 목표를 달성하기 위해 순차적 의사결정 문제를 상태, 행동, 보상, 그리고 상태전이 확률을 수학적으로 모델링 하여 나타낸 이산시간 확률(stochastic process) 과정을 마르코프 결정 프로세스(MDP)라 한다. 에이전트의 목표는 MDP에서 수차례의 시행착오를 통해 문제를 해결하는 것이다. 학습의 방법 중 정책 기반 강화학습이란 에이전트가 직접 정책 파라미터 공간을 탐색하여 목적함수를 최대화하는 정책을 유도하는 방식이다. 최근 강화학습에 딥러닝의 심층 신경망을 이용해 파라미터를 업데이트하여 목적함수를 최대로 하는 방식인 심층 강화학습(DRL, deep reinforcement learning)이 제안되었다.[1] Fig. 3.1.은 심층 강화학습의 학습 과정을 보여준다.

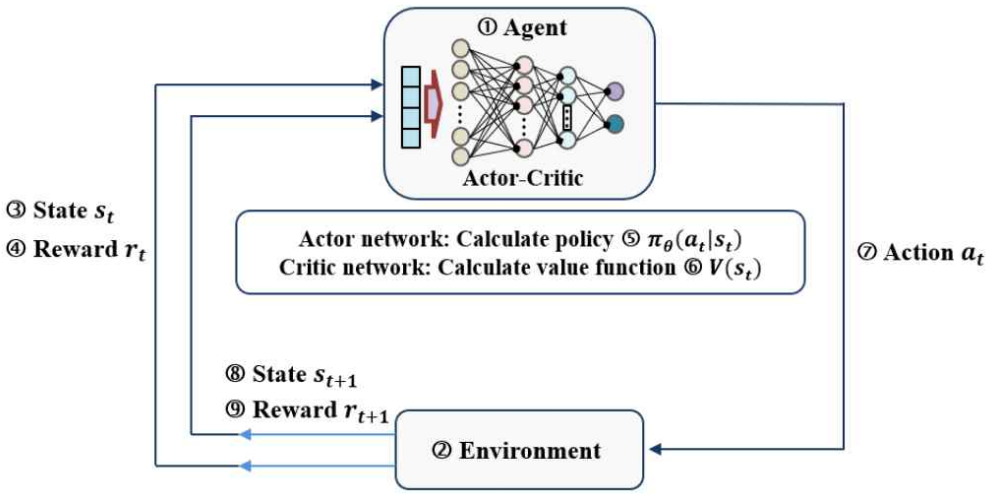

Fig. 3.1. The general DRL learning procedure[4]

학습 환경(Fig. 3.1-②)에서 의사결정자인 에이전트(Fig. 3.1-①)은 각 타임스텝마다 환경의 변화를 나타내는 상태 $s_t \in S$(Fig. 3.1-③)의 변화에 따라 보상 r_t(Fig. 3.1-④)를 받는다. 여기서, S는 환경 속에서 가능한 모든 상태의 집합이다. 보상은 문제를 해결하기 위한 중요한 요소이며, 에이전트는 정해진 종료 시점 동안 얻는 총 보상을 최대화 하는 것을 목표로 한다. 액터 신경망에서는 상태를 입력받아 현재 상태에서 어떤 행동 $a_t \in A(s_t)$(Fig. 3.1-⑦)을 할 것인가를 결정하는 정책 $\pi_\theta(a_t|s_t)$(Fig. 3.1-⑤)를 출력하고, 크리틱 신경망에서는 현재 상태 s_t로부터 얻을 수 있는 상태가치 $V(s_t)$(Fig. 3.1-⑥)을 출력한다. 여기서, $A(s_t)$는 에이전트가 현재 상태에서 선택할 수 있는 모든 행동의 집합이다. 에이전트가 심층 신경망으로부터

출력된 정책과 상태가치로부터 행동 a_t를 결정하게 되면 상태는 s_{t+1}(Fig. 3.1-⑧)이 되고 변화된 상태에 대한 보상 r_{t+1}(Fig. 3.1-⑨)를 얻는다. 학습 시작부터 종료 시점까지 에이전트가 탐색하고 보상을 받는 과정을 에피소드라 부르고 각각의 에피소드마다 심층 신경망의 가중치를 업데이트하여 문제의 해답을 찾는 액터-크리틱 방식의 대표적인 학습방법으로 PPO 알고리즘[16]이 있다.

3.1.1. PPO 알고리즘

근접 정책 최적화(PPO, proximal policy optimization)는 액터-크리틱 방식의 정책 기반 심층 강화학습 알고리즘이다. 제약조건을 2차 함수로 근사한 KL 발산(Kullback-Leibler divergency)을 이용해 선형화된 목적함수를 최대화 시키는 TPPO 알고리즘의 복잡한 계산을 단순화시키는 목적을 가지며, 정책의 크기를 일정 범위로 제한시켜 목적함수를 최대화 시키는 방법을 사용하기 때문에 안정적으로 정책 업데이트가 가능한 알고리즘이다.[16] PPO 알고리즘은 현재 정책과 이전 정책 사이의 차이를 줄이기 위해 식 (3.1)과 같이 클리핑(clipping)을 이용하였다.

$$clip(\eta_i(\theta),1-\epsilon,1+\epsilon)=\begin{matrix}1+\epsilon & \text{if, } \eta_i(\theta)\geq 1+\epsilon \\ 1-\epsilon & \text{if, } \eta_i(\theta)\leq 1-\epsilon \\ \eta_i & otherwise\end{matrix} \tag{3.1}$$

여기서, ϵ는 클리핑 하이퍼파라미터(hyperparameter)이며 $\eta_i=\frac{\pi(a_i|s_i)}{\pi_{old}(a_i|s_i)}$로 이전 정책에 대한 현재 정책의 비율이며 $[1-\epsilon,1+\epsilon]$ 범위로 제한된다.

PPO 알고리즘은 이전 정책을 통해 얻은 샘플을 이용하여 현재의 정책을 평가하는 on-policy이며 정책 $\pi(a_t,s_t)$를 계산하기 위해 액터 신경망은 상태로부터 가우시안 정책 확률밀도함수의 평균과 표준편차를 출력한다. 크리틱 신경망에서는 상태를 입력받아 상태가치를 계산한다. 크리틱 신경망의 파라미터를 ψ라 할 때 손실함수는 시간차 타깃과 상태가치 함수 $V_\psi(s_t)$의 평균 제곱 오차(MSE)를 이용한 아래의 식 (3.2)와 같이 정의된다.

$$L(\psi)=\frac{1}{2B}\sum_i(y_i-V_\psi(s_i))^2 \tag{3.2}$$

여기서, B는 미니배치 크기이고 y_i는 시간차 타깃으로 $y_i=r(s_t,a_t)+\gamma V_\psi(s_{t+1})$로 정의된다.

PPO 알고리즘에서 점진적인 정책 업데이트를 위해 클리핑이 도입된 대체(surrogate) 목적함수는 식 (3.3)과 같다.

$$L^{clip}(\theta)=E\left[\min\left\{\eta_t(\theta)A^{\pi_{\theta old}}(s_t,a_t),\, clip(\eta_t(\theta),1-\epsilon,1+\epsilon)A^{\pi_{\theta old}}(s_t,a_t)\right\}\right] \tag{3.3}$$

여기서, $A^{\pi_\theta}(s_t,a_t)$는 이점 함수(advantage function)로 시간 t에서 선택한 행동이 반환값의 기댓값에 미치는 영향의 척도를 뜻하며 무한 구간에서 식 (3.4)와 같다.

$$A^{(\infty)}(s_t,a_t)=\sum_{k=t}^{\infty}\gamma^{k-t}r(s_t,a_t)-V(s_t) \tag{3.4}$$

여기서, γ는 감가율이고 $V(s_t)$는 상태가치 함수로 t시간 이후부터 종료시점 까지 상태 s_t에서 정책 π로 얻어지는 총 보상의 기댓값이다. n-step 까지 계산한 어드밴티지는 식 (3.5)와 같이 정의된다.

$$A^{(n)}(s_t,a_t)=\sum_{k=t}^{t+n-1}\gamma^{k-t}r(s_t,a_t)+r^n V(s_{t+n})-V(s_t) \tag{3.5}$$

이점 함수 $A^{\pi_\theta}(s_t,a_t)$의 추정값의 편향과 분산을 줄이기 위해 식 (3.6)과 같이 n-step 어드밴티지 추정값에 가중치를 곱해 합산하는 어드밴티지 추정의 일반화(GAE, generalized advantage estimation)을 적용하였다.

$$GAE^t=\sum_{n=1}^{\infty}w_n A^{(n)}(s_t,a_t)=\sum_{k=t}^{\infty}(\gamma\lambda)^{k-t}\delta_k \tag{3.6}$$

여기서, δ_k는 1-step 시간차 오차로 $\delta_t=r(s_t,a_t)+\gamma V(s_{t+1})-V(s_t)$로 정의된다. λ는 [0,1]의 범위를 갖는 GAE parameter로 λ값을 조절하여 분산과 편향을 조절할 수 있다. λ이 0 또는 1인 경우 GAE^t는 식 (3.7)로 정의된다.

$$GAE^t=\begin{cases} r(s_t,a_t)+\gamma V_\psi(s_{t+1})-V_\psi(s_t)=A^{(1)}(s_t,a_t) & when\ \lambda=0 \\ \sum_{k=t}^{\infty}\gamma^{k-t}\delta_k=A^{(\infty)}(s_t,a_t) & when\ \lambda=1 \end{cases} \tag{3.7}$$

결국, PPO 알고리즘은 클리핑과 GAE를 도입함으로써 이전 정책과 현재 정책의 차이를 제한시키며 목적함수를 최대화 하는 식 (3.8)과 같은 최적화 문제로 정의할 수 있다.

$$\theta \leftarrow \arg max \sum_{t=0}^{\infty} E_{s_0 : a_t \sim p_{\theta old}(s_0 : a_t)} [L^{clip}(\theta)]$$
$$L^{clip}(\theta) = E_t \left[\min \left\{ \eta_t(\theta) GAE^t, clip(\eta_t(\theta), 1-\epsilon, 1+\epsilon) GAE^t \right\} \right] \quad (3.8)$$

3.2. 오프셋 데이터 최적 보간을 위한 DRL

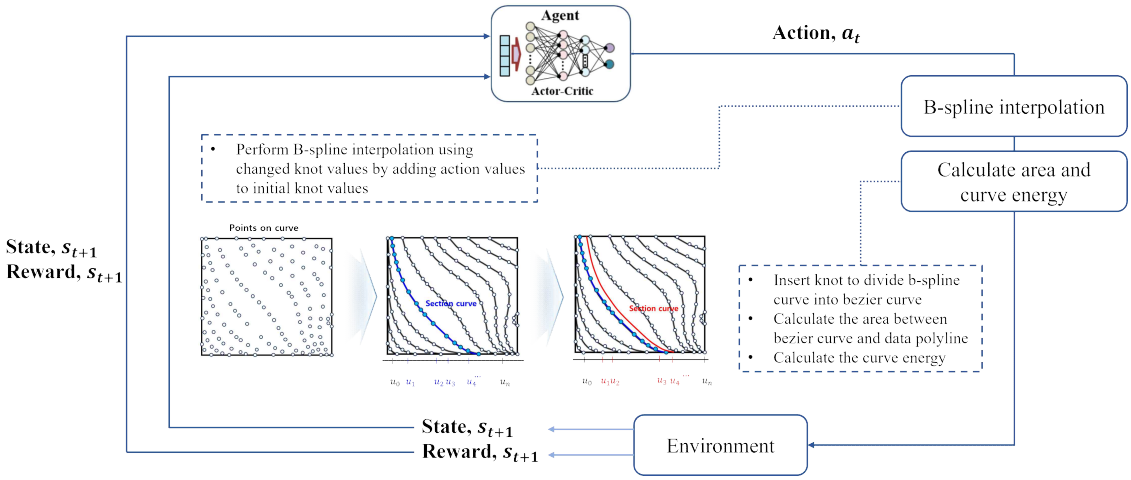

Fig. 3.2. DRL process for offset point interpolation

Fig. 3.2.는 심층 강화학습을 이용한 오프셋 데이터 최적 보간의 알고리즘 개념도이다. 알고리즘의 첫 번째 단계로, 매개변수화 방법 중 가장 정확하고 안정적인 방법이라 알려진 centripetal 매개변수화 방법을 이용하여 주어진 오프셋 데이터의 매듭 값을 계산한 후 에이전트가 관측할 수 있는 상태를 정의한다. 액터-크리틱 방식의 강화학습 알고리즘을 통해 계산된 행동($a \in R$)을 이용하여 매듭 값의 변환을 수행한다. 매듭 값을 변환하는 방법으로 상태로 정의한 매듭 값에 ±값인 행동을 더하였다. 변환된 매듭 값을 이용하여 B-spline 곡선 보간을 수행한 후, data polyline과 보간된 곡선 간의 넓이 값과 곡선의 에너지 값을 계산하여 에이전트에게 상태의 변화에 대한 보상을 준다. 에이전트는 누적 보상이 최대가 되는 정책을 찾는 것을 목표하며, 설정한 에피소드 수만큼 같은 과정을 반복한다. 3.2.절에서는 본 연구의 알고리즘 적용을 위한 상태, 행동, 보상의 수학적 모델링 과정을 기술한다.

3.2.1. 상태(State) 정의

상태는 알고리즘 내에서 환경의 변화를 나타내는 요소로 학습과정에서 중요한 부분이다. 액터-크리틱 방식의 강화학습 알고리즘에서 심층 신경망의 입력값으로 상태를 사용해 얻은 출력값을 이용해 행동을 결정하기 때문에 상태는 에이전트 행동에 영향을 받고 관측 가능한 변수로 정의해야한다. 본 연구에서는 보간을 위해 주어진 점 데이터을 이용하여 계산한 매듭 값을 상태로 정의하였다.

3.2.2. 행동(Action) 정의

에이전트의 행동은 이산(discrete) 행동과 연속(continuous) 행동으로 구분할 수 있는데, 상태의 변화를 고려하여 정교한 매듭 값의 변환을 수행하기 위해 실수 범위 내에서 연속적인 값을 선택하기 위해 연속 행동으로 정의하였다. centripetal 매개변수화 방법으로 계산한 초기 매듭 값은 오름차순으로 계산된다. 이후 학습과정에서 상태로 정의한 변수의 수에 맞게 계산된 액션 값을 각각의 매듭 값에 더하여 B-spline 곡선 보간을 수행하는데, 이 과정에서 i번째 매듭 값이 $i+1$번째 매듭 값보다 커지는 경우가 발생할 수 있다. 이런 문제를 해결하기 위해 행동의 최대 최소 범위를 Table. 3.1.과 같이 정의하였다.

Table. 3.1. Define the scope of action

Action range calculation order
Calculate the difference between the i^{th} knot value and $i+1^{th}$ knot value
Divide the difference by 2
Find minimum value of divided value
Divide the minimum value by set number of epochs, a
Set the action range to $[-a, a]$

위와 같은 방식으로 매듭 값에 연속 행동 a를 적용하면 Fig. 3.3.과 같이 변환된 매듭 값을 얻을 수 있다.

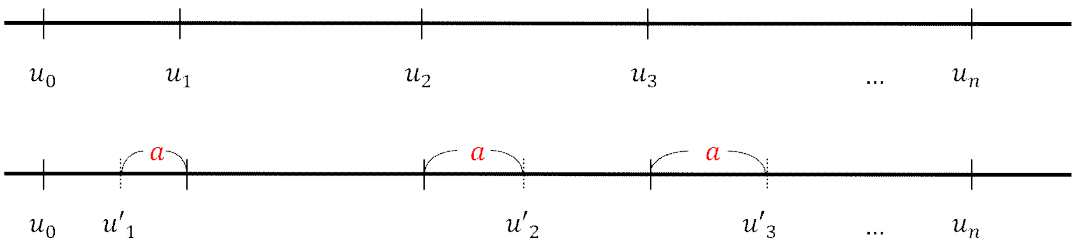

Fig. 3.3. 기존 매듭 값에 행동을 적용한 예

3.2.3. 보상(Reward) 정의

강화학습은 각각의 상태에서 에이전트가 행동을 선택했을 때 얻는 상태가치 $V(s_t)$가 최대화되는 방향으로 학습을 수행한다. 따라서 에이전트가 행동을 수행하였을 때 변환된 상태에 대해 주어지는 보상은 에이전트의 학습 방향을 결정짓는 중요한 요소 중 하나이다. 본 연구에서는 B-spline 곡선 보간 과정에서 매듭 값을 변화시켜 B-spline 곡선 보간 수행 후 data polyline과 보간된 곡선이 이루는 면적의 값

과 보간된 곡선의 에너지를 최소화하는 오프셋 데이터 최적 보간을 목표로 한다.

Data polyline과 보간된 곡선이 이루는 면적 값은 Fig. 3.4.와 같이 계산된다. 예로 4개의 곡선 위의 점($\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$)이 주어진 경우, 보간된 B-spline 곡선을 곡선 위의 점을 이용하여 n개의 3차 Bézier 커브로 나눈다. 각 Bézier 커브의 첫 점과 끝 점은 처음 주어진 곡선 위의 점($\mathbf{p}_i, \mathbf{p}_{i+1}$)이므로 해당 점의 x좌표와 y좌표를 이용하여 data polyline을 생성한 후, data polyline과 x축이 이루는 각 θ를 이용하여 3차 Bézier 곡선 상의 점과 제어점을 θ만큼 회전변환 한다. 회전변환 시킨 각각의 3차 Bézier 곡선을 x축에 대하여 적분한 값을 더하여 B-spline 곡선과 data polyline간의 넓이로 정의한다.

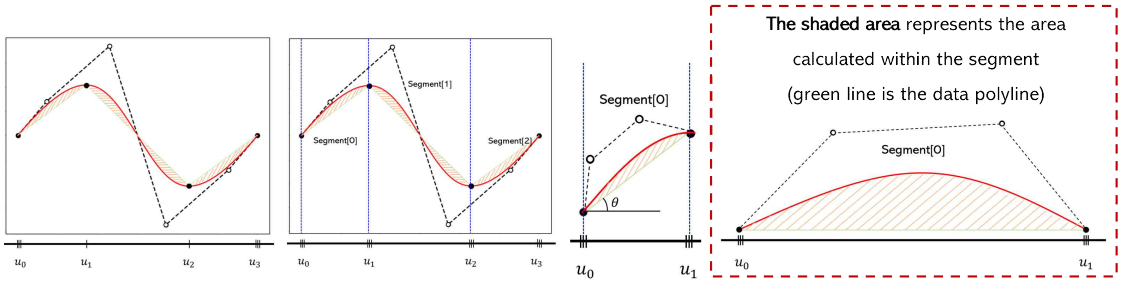

Fig. 3.4. data polyline과 보간된 곡선 간 넓이를 구하는 예

다음으로 보간된 곡선의 부드러운 정도를 나타내는 곡선의 에너지는 2.3절에서 설명한 식(2.7)을 이용하여 구한다. 곡선의 에너지는 곡률과 호의 길이로 정의되므로 부드럽지 않은 곡선, 즉 굽은 구간이 많은 곡선일수록 곡선의 에너지는 크게 계산된다.

$$r = \begin{cases} \text{if original energy} > \text{new energy}: \\ \quad \text{if original area} > \text{new area}: 0 \\ \quad \text{else}: -10 \\ \text{else}: \\ \quad \text{if original area} > \text{new area}: -20 \\ \quad \text{else}: -30 \end{cases} \tag{3.9}$$

본 연구에서는 초기 매듭 값으로 보간한 곡선을 이용하여 구한 면적 값과 에너지 값을 보상 값의 기준으로 설정하였다. 최종적으로 학습에 사용된 보상 r은 식(3.9)와 같다. 곡선의 에너지가 작을수록 부드럽게 보간된 곡선이므로 data polyline과 보간된 곡선과의 면적 값이 작아지는 경향이 있다.

에이전트는 각각의 시간스텝 t에서 매듭 값을 변환한 후 변환된 매듭 값을 이용하여 보간한 곡선의 에너지 값과 넓이 값을 계산한 후, 기준으로 설정한 값과 비교하여 식(3.9)와 같이 보상을 받게 된다.

3.3. DRL 알고리즘 적용 결과 및 분석

3.3.절에서는 DRL을 B-spline 곡선 보간 문제에 적용한 결과와 GA 알고리즘을 적용한 결과를 서술한다.

3.3.1. PPO 알고리즘 적용

Table. 3.2. The scheme of ppo algorithm iteration[16]

Algorithm. Proximal Policy Optimization Algorithm
for iteration = 1, 2, ..., do
for iteration=1, 2, ..., N do
run policy $\pi_{\theta old}$ in environment for T timesteps
Compute advantage estimates $A^{\pi_\theta}(s_1,a_1)$, $A^{\pi_\theta}(s_2,a_2)$,..$A^{\pi_\theta}(s_T,a_T)$
end for
Optimize surrogate object function L wrt θ, with K epochs and mini batch size $M \leq NT$
$\theta_{old} \leftarrow \theta$
end for

PPO 알고리즘은 Table. 3.2.와 같이 작동한다. 오프셋 데이터 최적 보간 프로세스에 PPO 알고리즘을 적용해 보면, 액터-크리틱 신경망의 파라미터 θ, ψ를 초기화한 후 현재 상태로부터 액터 신경망에서 이전 정책 $\pi_{\theta old}$를 가우시안으로 가정하고 평균과 표준편차를 계산한 후 로그-정책 확률밀도함수를 계산한다. PPO 알고리즘은 확률론적 정책 그래디언트 방식으로 액터-크리틱 신경망의 구조는 Fig. 3.5.와 같다.

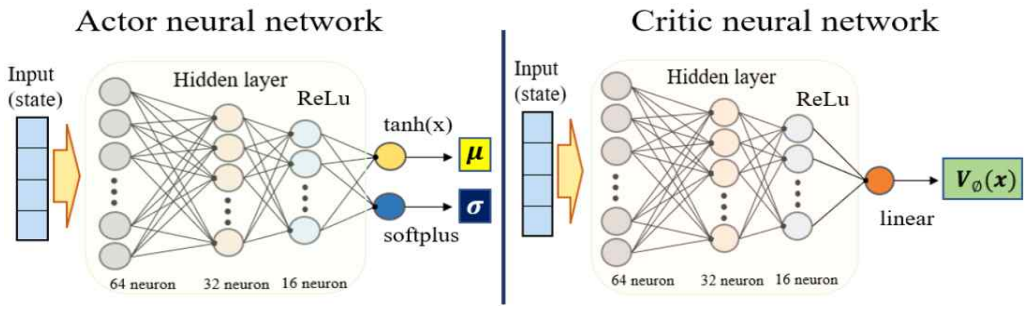

Fig. 3.5. Structure of the actor-critic neural network in PPO algorithm

정책으로부터 결정된 행동을 수행하여 보상 r_i와 다음 상태 s_{t+1}를 계산한다. 이후 시간차 타깃 y_i와 이점 함수 $A^{\pi_\theta}(s_i,a_i)$를 계산한다. 시간스텝 T만큼 샘플링된 묶음 $(s_i, a_i, y_i, \log\pi_{\theta old}(a_i|s_i))$를 배치에 저장한다. 액터-크리틱 신경망의 학습 에포크

만큼 다음 과정을 반복한다. 배치로부터 미니배치 크기 B만큼 데이터를 추출하여 크리틱 신경망의 손실함수를 계산한다. 확률적 경사하강법을 이용해 크리틱 신경망의 가중치 ψ를 업데이트 한다. 이전 정책에 대한 현재 정책의 비율 $r_t(\theta)$를 계산한 후 대체 목적함수의 그래디언트 $\nabla L^{clip}(\theta)$를 이용해 액터 신경망의 가중치 θ를 업데이트 한다.

액터-크리틱 신경망의 옵티마이저로는 Adam optimizer를 사용하였다. 오프셋 데이터 최적 보간 전, 간단한 예제의 점 데이터를 이용하여 B-spline 곡선 보간을 수행하였다. Fig. 3.6.은 예시 점 데이터를 centripetal 매개변수화 방법과 PPO 알고리즘을 이용하여 B-spline 곡선 최적 보간을 구현한 곡선이다. centripetal 매개변수화 방법을 이용하여 보간된 곡선의 에너지 값은 231.211이고 넓이 값은 144.47이며 PPO 알고리즘을 이용하여 보간한 곡선의 에너지 값은 219.704, 넓이 값은 68.539로 각각 4.98%, 52.56% 감소하였다.

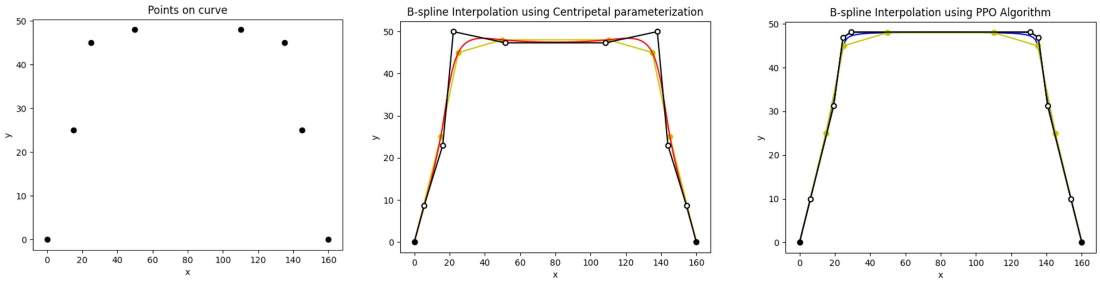

Fig. 3.6. 기존의 매개변수화 방법과 PPO 알고리즘을 이용하여 보간한 예

3.4. 최적화 알고리즘 결과 비교

본 절에서는 유전학 개념을 기반으로 주어진 문제의 잠재해 사이에 자연선택과 유전법칙을 적용하여 매개변수 값을 제한 범위 내에서 조절하여 주어진 목적함수를 최대화 또는 최소화시키는 해를 찾는 유전 알고리즘에 B-spline 곡선 최적 보간 문제를 정의하여 얻어지는 결과와 심층 강화학습을 이용한 B-spline 곡선 최적 보간 결과를 비교한다. 유전 알고리즘으로는 NSGA-II(non-dominated sorting genetic algorithm)[9]를 사용하였다.

Fig. 3.7.은 유전 알고리즘의 흐름도 나타낸다. 유전 알고리즘의 초기 모집단을 생성할 때 실수 값을 갖는 유전자(gene)를 난수로 발생 시켜 2개의 유전자를 갖는 염색체(chromosome)를 구성하였다. 강화학습의 경우 상태의 변화를 관측하며 각각의 상태에서의 행동을 수행하여 주어진 점 데이터 최적 보간을 진행하는 반면 유전 알고리즘은 무작위로 생성된 난수의 조합을 통해 적합도 평가와 진화 과정을 반복하여 얻은 최적해로 한 번의 매듭 값 변환을 통해 최적 매듭 값을 찾는 차이가 있다.

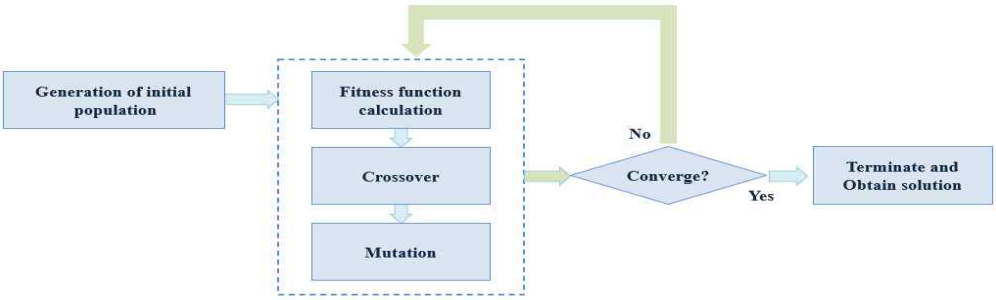

Fig. 3.7. Flow Chart of Genetic Algorithm

Fig. 3.8.은 예시 점 데이터를 centripetal 매개변수화 방법과 GA 알고리즘을 이용하여 B-spline 곡선 최적 보간을 구현한 곡선이다. GA 알고리즘을 이용하여 보간한 곡선의 에너지 값은 219.809, 넓이 값은 70.161로 각각 4.93%, 51.44% 감소하였는 것을 알 수 있다. 두 알고리즘의 에너지 감소율 차이는 0.05%, 넓이 감소율 차이는 1.12%이다.

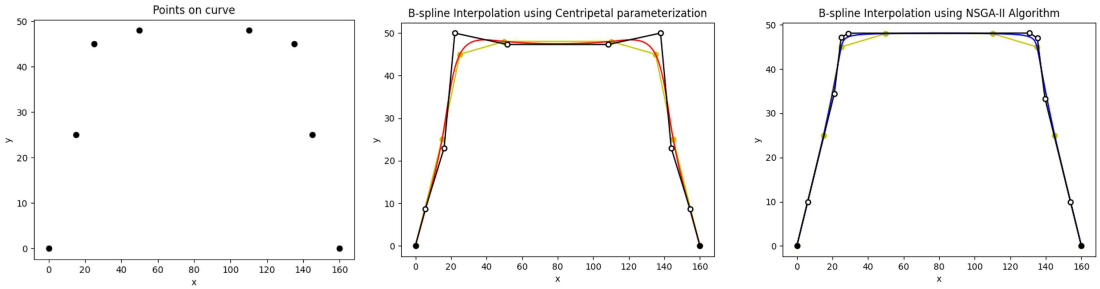

Fig. 3.8. 기존의 매개변수화 방법과 GA 알고리즘을 이용하여 보간한 예

Fig. 3.9.는 다른 예시 점 데이터를 centripetal 매개변수화 방법과 PPO 알고리즘, GA 알고리즘을 이용하여 B-spline 곡선 최적 보간을 구현한 곡선이다. centripetal 매개변수화 방법을 이용하여 보간된 곡선의 에너지 값은 444.44이고 넓이 값은 246.82이며 PPO 알고리즘을 이용하여 보간한 곡선의 에너지 값은 441.15, 넓이 값은 240.18로 각각 0.74%, 2.69% 감소하였다. GA 알고리즘을 이용하여 보간한 곡선의 에너지 값은 432.95, 넓이 값은 305.90로 에너지 값은 2.659% 감소한 반면, 넓이 값은 23.94% 증가하였다. GA 알고리즘을 이용하였을 경우, 에너지 값은 PPO 알고리즘을 이용하여 구현한 결과보다 많이 감소하였지만 넓이 값을 살펴볼 때 크게 증가하였다. 해당 결과를 살펴보았을 때, GA 알고리즘을 이용한 모델은 최적화가 완벽하게 진행되지 않았을 가능성이 있다.

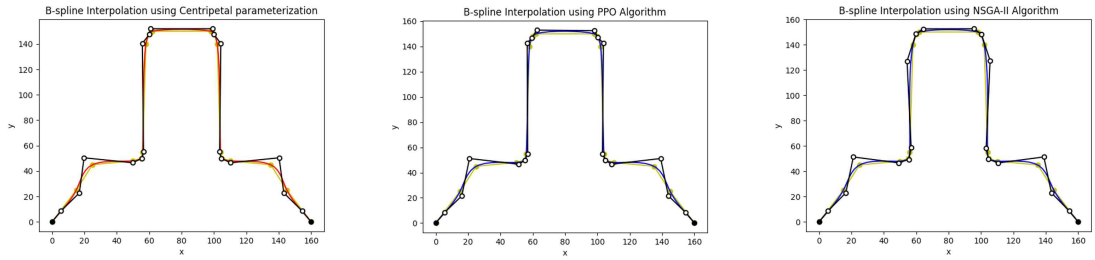

Fig. 3.9. 기존의 매개변수화 방법과 PPO알고리즘, GA 알고리즘을 이용하여 보간한 예

3.5. 오프셋 데이터 적용

본 절에서는 제안하는 모델에 선형을 나타내는 오프셋 데이터를 적용하여 각 섹션별로 구현한 단면 곡선과, 단면 곡선들을 이용해 생성한 body plan에 대해 서술한다. 본 연구에서는 알고리즘 적용 선박으로 KVLCC2[17]을 선정하였다. KVLCC2 선박의 오프셋 데이터를 적용하여 20개의 단면 곡선에 대해 학습을 진행해 최적의 매듭 값을 계산하였다.

3.5.1. 오프셋 데이터를 PPO 알고리즘에 적용한 결과

Fig. 3.10.은 KVLCC2 선박의 1번 스테이션의 오프셋 데이터를 이은 data polyline과 해당 오프셋 데이터를 기존의 매개변수화 방법, PPO 알고리즘에 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 38.637, 넓이 값은 6.608이고 PPO 알고리즘을 이용했을 때는 각각 38.385, 5.549으로 0.65%, 16.03% 감소하였다. Fig. 3.11은 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

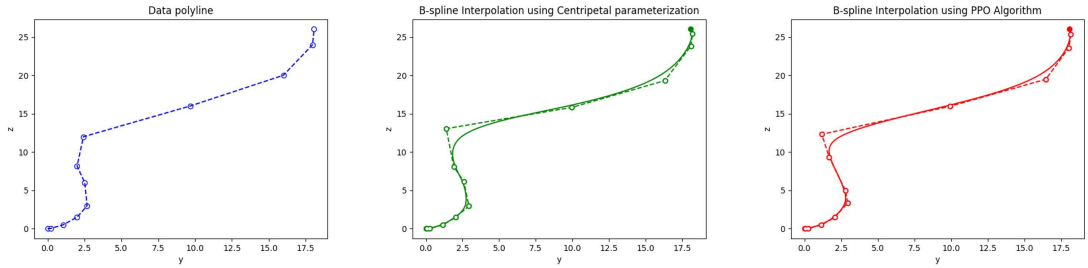

Fig. 3.10. KVLCC2 선박의 1번 station 오프셋 데이터를 적용하여 보간한 예

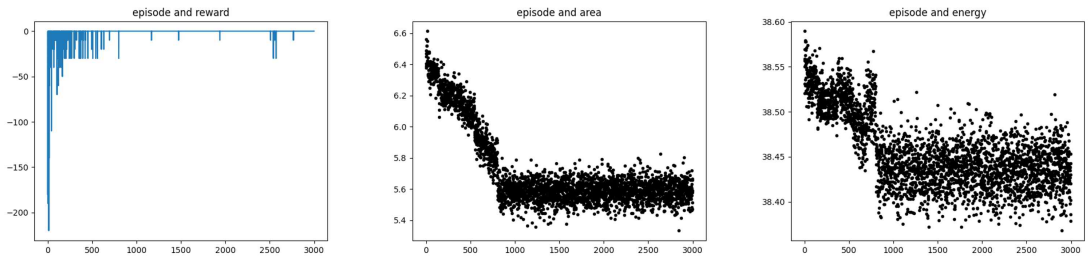

Fig. 3.11. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.12.는 KVLCC2 선박의 2번 스테이션의 오프셋 데이터를 이은 data polyline과 해당 오프셋 데이터를 기존의 매개변수화 방법, PPO 알고리즘에 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지

값은 37.277, 넓이 값은 4.824이고 PPO 알고리즘을 이용했을 때는 각각 37.251, 4.683으로 0.07%, 2.92% 감소하였다. Fig. 3.13은 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

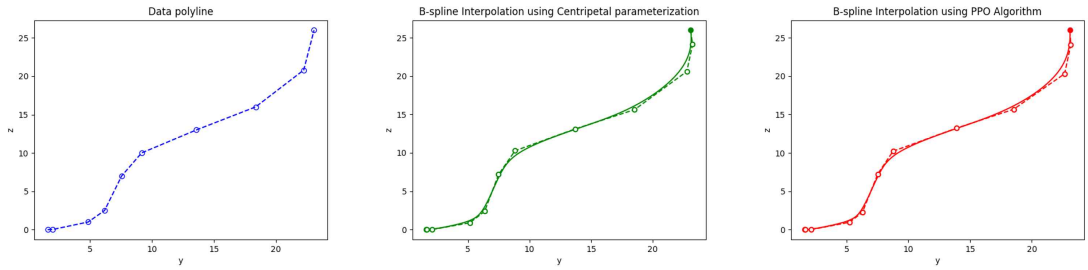

Fig. 3.12. KVLCC2 선박의 2번 station 오프셋 데이터를 적용하여 보간한 예

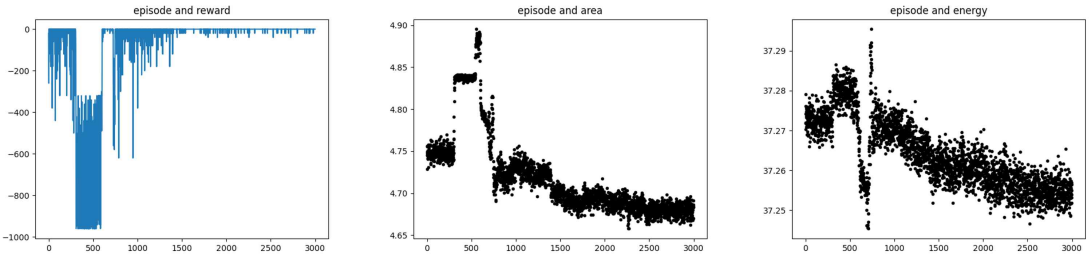

Fig. 3.13. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.14.는 KVLCC2 선박의 3번 스테이션의 오프셋 데이터를 이은 data polyline과 해당 오프셋 데이터를 기존의 매개변수화 방법, PPO 알고리즘에 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 37.199, 넓이 값은 3.499이고 PPO 알고리즘을 이용했을 때는 각각 37.158, 3.417으로 0.11%, 2.34% 감소하였다. Fig. 3.15는 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

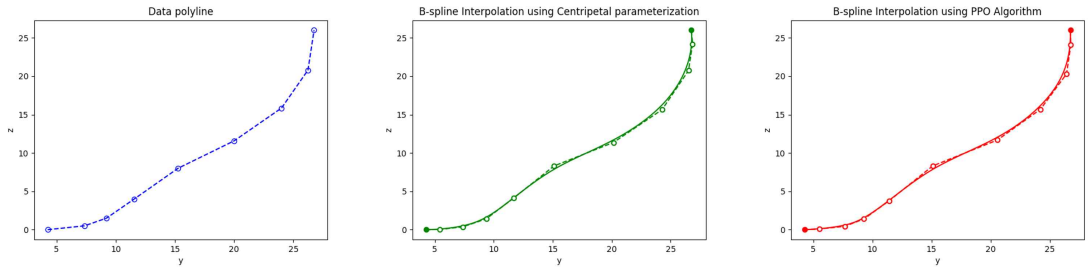

Fig. 3.14. KVLCC2 선박의 3번 station 오프셋 데이터를 적용하여 보간한 예

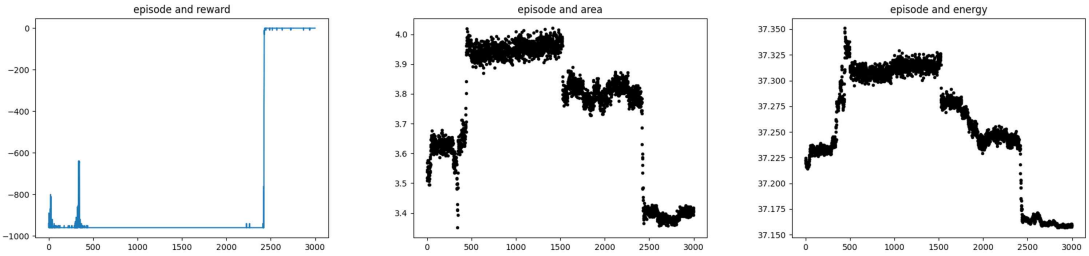

Fig. 3.15. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.16.은 KVLCC2 선박의 4번 스테이션의 오프셋 데이터를 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 39.346, 넓이 값은 3.452이고 PPO 알고리즘을 이용했을 때는 각각 39.3, 3.385으로 0.12%, 1.94% 감소하였다. Fig. 3.17.은 해당 오프셋 데이터를 이용하여 학습할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

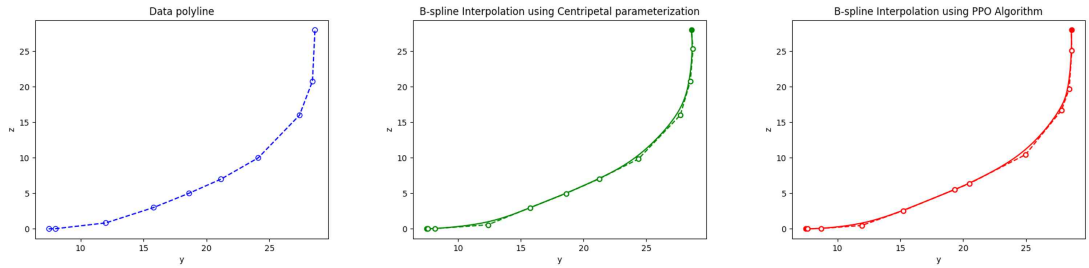

Fig. 3.16. KVLCC2 선박의 4번 station 오프셋 데이터를 적용하여 보간한 예

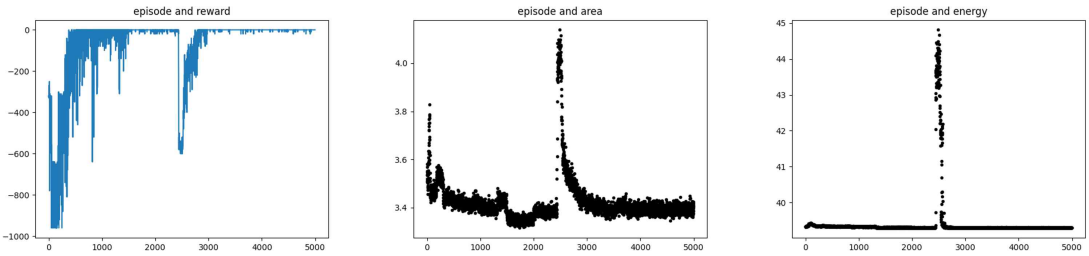

Fig. 3.17. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.18.은 KVLCC2 선박의 5번 스테이션의 오프셋 데이터를 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 38.752, 넓이 값은 3.218이고 PPO 알고리즘을 이용했을 때는 각각 38.729, 3.192으로 0.06%, 0.81% 감소하였다. Fig. 3.19.는 해당 오프셋 데이터를 이용하여 학습할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

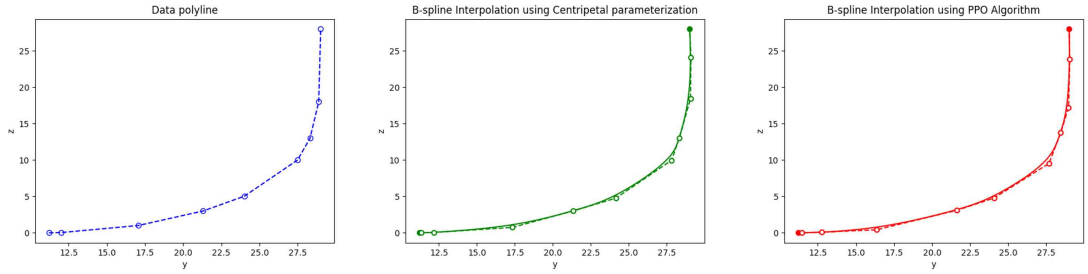

Fig. 3.18. KVLCC2 선박의 5번 station 오프셋 데이터를 적용하여 보간한 예

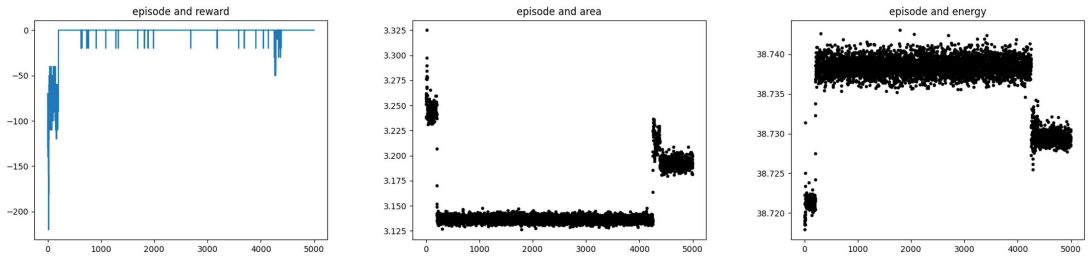

Fig. 3.19. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.20.은 KVLCC2 선박의 8, 9, 11, 12, 13, 14번 스테이션의 오프셋 데이터를 적용하여 단면 곡선을 생성한 그림이다. 8, 9, 11, 12, 13, 14번 스테이션의 오프셋 데이터 값이 같아 6개의 스테이션에 대해 한번에 학습을 진행하였다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 3.79954, 넓이 값은 0.045이고 PPO 알고리즘을 이용했을 때는 각각 3.79919, 0.042으로 0.01%, 6.67% 감소하였다. Fig. 3.21은 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

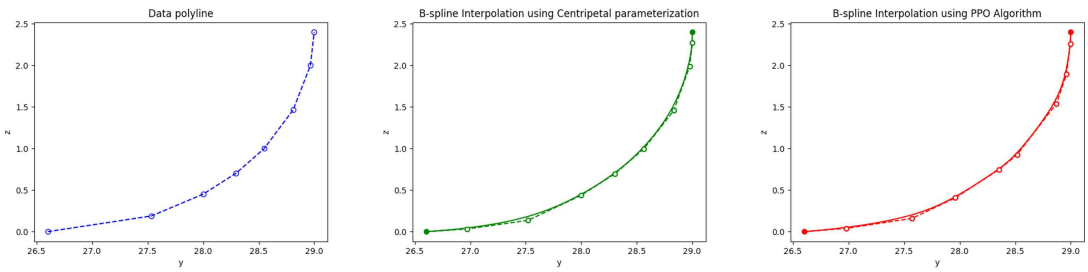

Fig. 3.20. KVLCC2 선박의 8, 9, 11, 12, 13, 14번 station 오프셋 데이터를 적용하여 보간한 예

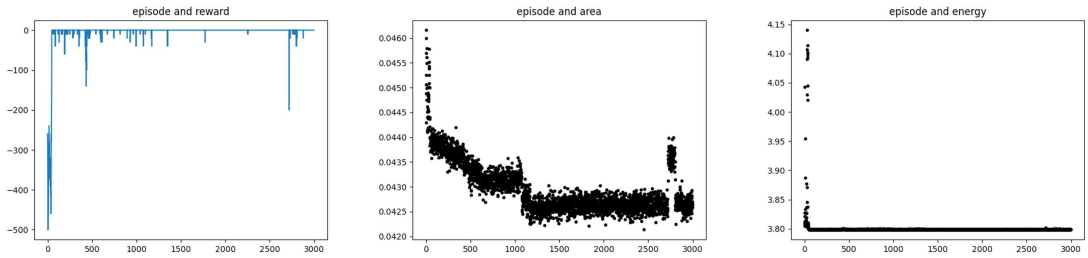

Fig. 3.21. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.22.는 KVLCC2 선박의 15번 station 오프셋 데이터를 적용하여 단면 곡선을 생성한 그림이다. 기존의 매개변수화 방법을 이용했을 때 에너지 값은 7.077, 넓이 값은 0.147이고 PPO 알고리즘을 이용했을 때는 각각 7.07, 0.146으로 0.1%, 0.68% 감소하였다. Fig. 3.23.은 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다.

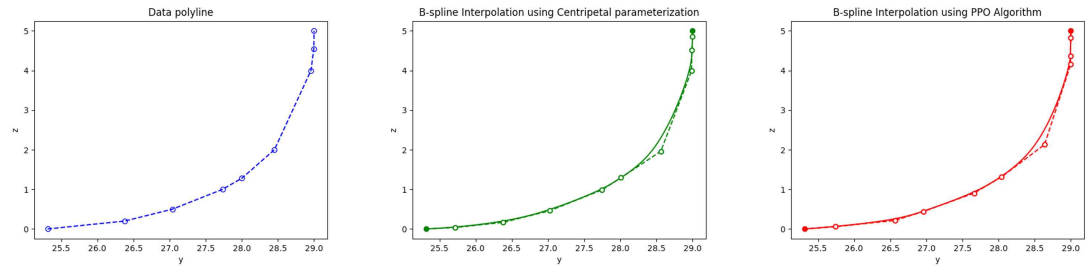

Fig. 3.22. KVLCC2 선박의 15번 station 오프셋 데이터를 적용하여 보간한 예

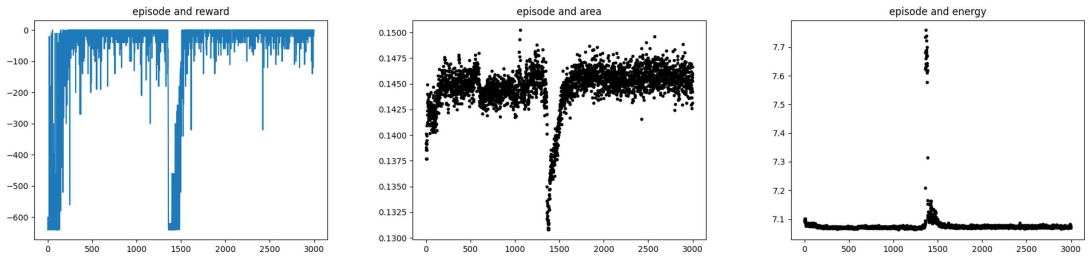

Fig. 3.23. 에피소드에 따른 보상 값과 목적함수 값

Fig. 3.24는 기존의 매개변수화 방법을 활용하여 생성된 body plan이고 Fig. 3.25는 PPO 알고리즘을 사용하여 학습한 모델에 오프셋 데이터를 적용하여 생성된 단면 곡선을 보여주고, 이를 시각화한 body plan이다. 각 단면 곡선 별 감소율이 크지 않기 때문에, 해당 그림을 비교할 때, 직관적으로 차이가 없을 수 있다. 하지만 학습 결과를 살펴보았을 때, 설정한 목적함수가 감소하였음을 확인할 수 있으므로, 학습된 모델을 이용하여 선형을 생성할 때 기존의 방법을 이용하여 생성할 때보다 더 부드러운 선형을 생성할 수 있다.

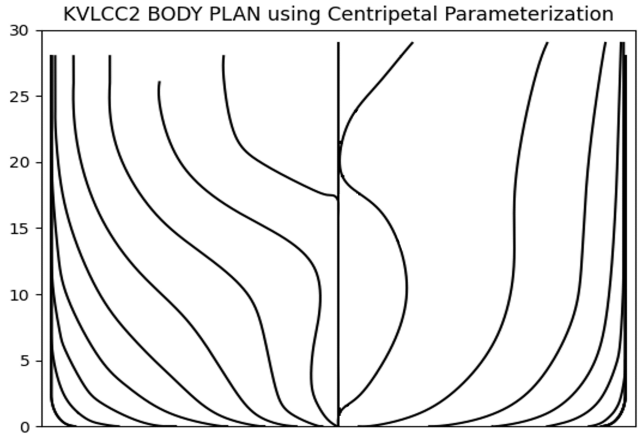

Fig. 3.24. 기존 매개변수화 방법을 이용하여 생성한 body plan

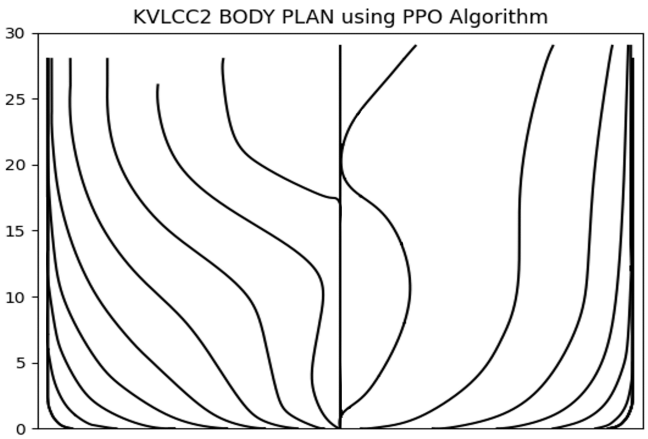

Fig. 3.25. PPO 알고리즘을 이용하여 생성한 body plan

Fig. 3.26.은 본 연구에서 제안하는 모델에서 초기 상태를 centripetal 매개변수화 방법이 아닌, uniform 매개변수화 방법을 적용하여 3번 스테이션의 오프셋 데이터를 적용하여 단면곡선을 생성한 그림이다. Fig. 3.27.은 해당 오프셋 데이터를 이용하여 학습 할 때 에피소드에 따른 보상 값과 목적함수를 나타낸 그림이다. Uniform 매개변수화 방법을 이용했을 때 에너지 값은 37.73, 넓이 값은 3.54이고 PPO 알고리즘을 이용했을 때는 각각 37.28, 3.43으로 1.19%, 3.11% 감소하였다.

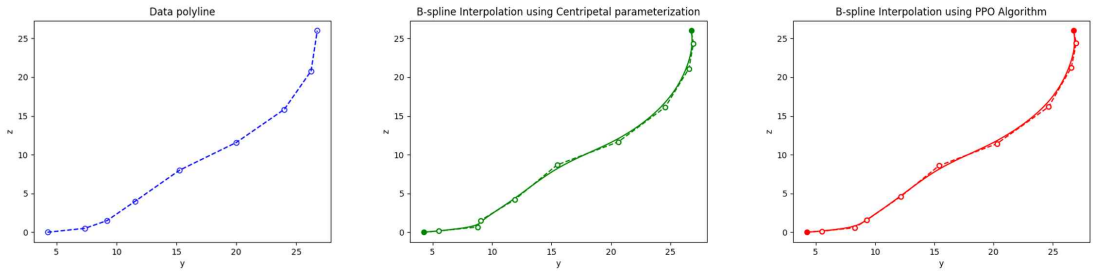

Fig. 3.26. 에피소드에 따른 보상 값과 목적함수 값

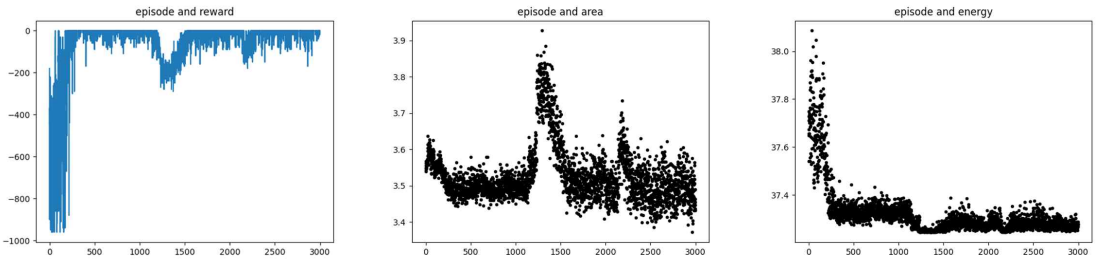

Fig. 3.27. 에피소드에 따른 보상 값과 목적함수 값

초기 상태를 centripetal 매개변수화 방법을 이용하였을 때 에너지 값과 넓이 값은 0.11%, 2.34% 감소하였고 uniform 매개변수화 방법을 이용하였을 때 에너지 값과 넓이 값은 1.19%, 3.11% 감소하였다. 해당 결과를 통해, Uniform 매개변수화 방법을 이용하여 초기 상태를 정의한 경우와 centripetal 매개변수화 방법을 이용하여 초기 상태를 정의한 경우, 최적해를 찾아가는 변화량이 다르다는 것을 알 수 있다, 또한, uniform 매개변수화 방법으로 초기 상태를 정의하여 학습한 결과를 토대로, 모델의 성능을 확인할 수 있다.

3.6. 강화학습 모델의 적용

본 연구에서는 PPO 알고리즘을 이용하여 오프셋 데이터 최적 보간 모델을 구현하였다. 본 절에서는 강화학습 모델의 실용성을 검증하기 위해, KVLCC2 선박의 오프셋 데이터로부터 학습이 완료된 강화학습 모델을 이용하여 138K LNG선(KLNG)에 오프셋 데이터 최적 보간을 수행하였다. 강화학습 환경에서 에이전트는 변환된 매듭 값에 대하여 오프셋 데이터 최적 보간을 수행하며 해당 매듭 값으로 보간된 곡선에 대해 목적함수 값에 따라 보상 값을 받는다. 누적 보상 값이 최대가 되었을 때 오프셋 데이터 최적 보간이 완료되었다고 판단한다. Fig. 3.28.은 KLNG선에 학습된 모델을 적용한 예이다. 구현 결과, 에너지 값과 넓이 값은 0.17%, 4.21% 감소하였다.

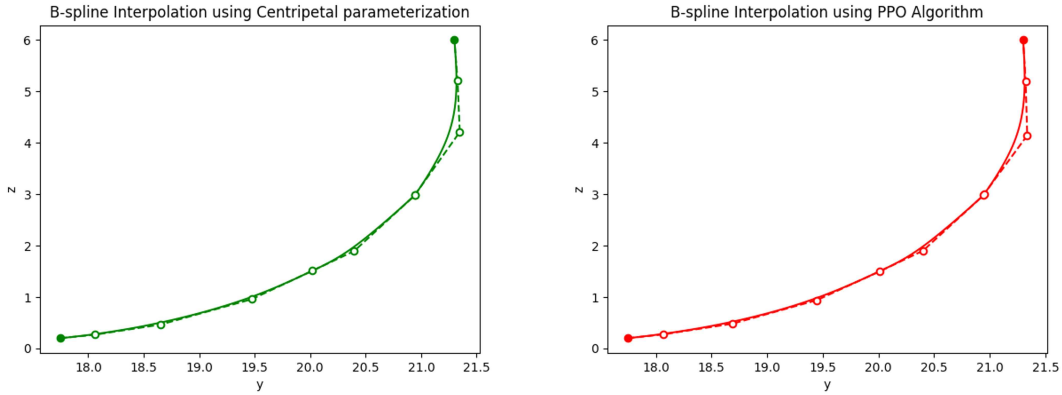

Fig. 3.28. 학습된 모델을 다른 선종에 적용한 예

학습된 모델을 적용하였을 때 5.41s의 시간 동안 오프셋 데이터 최적 보간을 수행하였다. 이미 학습된 모델을 적용할 경우, 처음 학습할 때 보다 시간이 매우 단축되지만, 학습된 모델의 오프셋 데이터 수와 학습할 모델의 오프셋 데이터 수가 같아야 학습이 가능하다는 단점이 존재한다.

제4장 결론 및 고찰

본 연구에서는 선형 생성에 사용되는 오프셋 데이터를 기반으로 하는 B-spline 곡선 보간 과정에서, 동일한 오프셋 데이터일지라도 매듭 값에 따라 보간 결과가 달라지는 문제를 해결하기 위해 심층 강화학습을 활용한 오프셋 데이터 최적 보간 방법을 제안하였다. 오프셋 데이터 최적 보간을 수행하는 에이전트는 centripetal 매개변수화 방법을 이용하여 초기 매듭 값을 계산해 상태를 정의하고 초기값에 액션 값을 더하여 매듭 값을 조정한 후 B-spline 곡선 보간을 수행하였다. 이후 주어진 점 데이터를 이은 data polyline과 보간된 곡선이 이루는 면적 값 및 보간된 곡선의 에너지 값에 대해 보상을 받도록 강화학습 문제를 정의하였다.

기존의 공학적 최적화 문제에 대표적으로 사용되고 있는 오프셋 데이터 최적 보간 문제에 전역 최적화 기법인 유전 알고리즘(NSGA-II)을 적용하여 PPO 알고리즘을 이용하여 학습한 결과와 비교 분석하였다.

강화학습을 사용하여 구현한 모델은 이미 학습된 가중치를 활용하여 다른 선박의 오프셋 데이터에 대해 바로 최적 해를 구할 수 있는 반면 유전 알고리즘을 사용할 경우 매번 문제를 재정의하고 탐색과정을 반복해야 하는 단점이 존재한다. 따라서 이전 정책과 현재 정책의 차이를 제한하며 점진적으로 정책을 업데이트하는 PPO 알고리즘이 오프셋 데이터 최적 보간 모델로 적합하다고 판단하였다.

PPO 알고리즘을 오프셋 데이터에 적용하여 오프셋 데이터 최적 보간 모델을 구현하였다. 구현 결과 기존의 매개변수화 방법을 이용하였을 때보다 PPO 알고리즘을 이용하여 학습한 모델을 적용했을 때 더 우세한 결과를 나타내었다. 또한, 초기 상태를 centripetal 매개변수화 방법과 uniform 매개변수화 방법을 이용하여 정의했을 때의 학습 결과를 비교 분석하였다. 초기 값의 차이로 인해 최적 해를 찾아가는 과정이 서로 다르므로 변화량 또한 다르다. 때문에 최적 해와 상대적으로 멀리 떨어져 있는 uniform 매개변수화 방법을 사용했을 때 목적 함수의 감소율이 더 크다.

학습된 모델을 다른 선종의 오프셋 데이터에 적용하여 학습하였다. 학습 결과, 최적 해를 계산하는데 5.41s 소요되며 목적 함수로 설정한 에너지 값은 0.17%, 넓이 값은 4.21% 감소하였다. 하지만 학습된 모델의 오프셋 데이터의 수와 새로운 선종의 오프셋 데이터의 수가 같아야 학습 할 수 있다는 단점이 존재한다.

향후 연구로 강화학습을 이용하여 모델을 학습할 때, 각 매듭 값의 변화 가능한 범위를 개별적으로 설정하여 학습한다면, 본 논문에서 제안하는 강화학습 모델보다 더 정교한 모델을 구현할 수 있을 것으로 기대된다.

참고문헌

[1] Sutton, Richard S., and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.
[2] Farin, Gerald E. Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann, 2002.
[3] Piegl, Les, and Wayne Tiller. The NURBS book. Springer Science & Business Media, 2012.
[4] Lee, Eugene TY. "Choosing nodes in parametric curve interpolation." Computer-Aided Design 21.6 (1989): 363-370.
[5] Nielson, Gregory M., and Thomas A. Foley. "A survey of applications of an affine invariant norm." Mathematical methods in computer aided geometric design. Academic Press, 1989. 445-467.
[6] Everitt, W. Norrie, and Hubert Kalf. "The Bessel differential equation and the Hankel transform." Journal of Computational and applied Mathematics 208.1 (2007): 3-19.
[7] Horn, Berthold KP. "The curve of least energy." ACM Transactions on Mathematical Software (TOMS) 9.4 (1983): 441-460.
[8] Fang, Lian, and David C. Gossard. "Multidimensional curve fitting to unorganized data points by nonlinear minimization." Computer-Aided Design 27.1 (1995): 48-58.
[9] Goldberg, David E. "Genetic Algorithms in Search." Optimization, Machine Learning (1989).
[10] Konak, Abdullah, David W. Coit, and Alice E. Smith. "Multi-objective optimization using genetic algorithms: A tutorial." Reliability engineering & system safety 91.9 (2006): 992-1007.
[11] Kakade, Sham M. "A natural policy gradient." Advances in neural information processing systems 14 (2001).
[12] Thomas, Philip. "Bias in natural actor-critic algorithms." International conference on machine learning. PMLR, 2014.
[13] Park S.S., 2021. “수학으로 풀어보는 강화학습 원리와 알고리즘”, Paju-si, WIKIBOOKS, pp. 91-104.
[14] Sutton, Richard S., et al. "Policy gradient methods for reinforcement learning with function approximation." Advances in neural information processing systems 12 (1999).
[15] Greensmith, Evan, Peter L. Bartlett, and Jonathan Baxter. "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning." Journal of Machine Learning Research 5.9 (2004).

[16] Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).
[17] MOERI Tanker KVLCC2, http://www.simman2008. dk/KVLCC/KVLCC2/kvlcc2_geometry.html.
[18] Hu, Liangchen, and Wensheng Zhang. "NSGA-II approach for proper choice of nodes and knots in B-spline curve interpolation." Computer-Aided Design 127 (2020): 102885.

부록

본 논문에서 기존의 최적화 알고리즘인 유전 알고리즘(Genetic Algorithm, GA)을 예제 데이터에 적용하였다.

– 유전 알고리즘

유전알고리즘이란 Holland(1975)에 의해 유전학 개념을 기반으로 제안된 최적화 기법으로, 주어진 문제의 목적함수를 최대화 또는 최소화하는 해를 찾기 위한 전역 최적화 방법 중 하나이다.[9] 자연계의 진화 과정을 모방한 확률적 탐색 방법으로, 다양한 해를 생성하고 성능을 평가하여 진화시킨다. 이를 문제에 적용하기 위해서는 목적 함수에 의해 정의된 문제의 해 공간을 이진수로 표현한 염색체(chromosome)를 가지고 있는 개체(population)들의 모집단을 생성해야 한다. 이 모집단은 알고리즘의 진화과정에서 계속해서 변경되며, 최적의 해를 찾는 데 사용된다. Fig. 1.는 유전 알고리즘에서 모집단의 구성 요소를 보여준다. 모집단은 다양한 개체로 구성되며, 각 개체는 문제 공간에서 특정 위치에 해당하는 염색체를 갖고 있다. 이진수로 표현된 염색체는 해의 후보는 나타내며, 이러한 해의 후보군을 평가하고 진화시킴으로써 최적의 해를 도출한다.

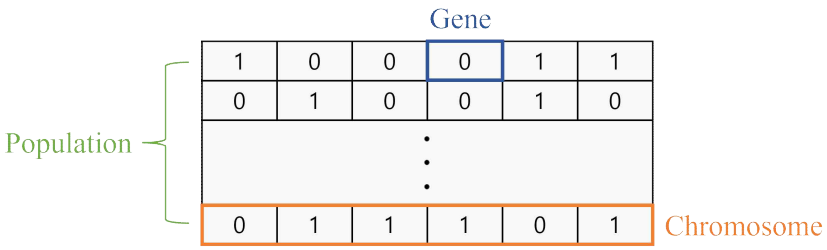

Fig. 1. 유전알고리즘 모집단의 구성 요소

유전 알고리즘에서 염색체는 유전자(gene)로 구성되며 해를 의미하는 X는 개체 또는 염색체라 정의한다. 유전 알고리즘은 초기 무작위로 주어진 개체들을 이용하여 선택(selection), 교배(crossover), 돌연변이(mutation) 세 개의 연산자를 통해 반복과 수혐 과정을 거쳐 개체를 진화시키는 방식으로 새로운 자손을 생성한다. Fig. 2.은 유전 알고리즘의 기본 도식도이다.

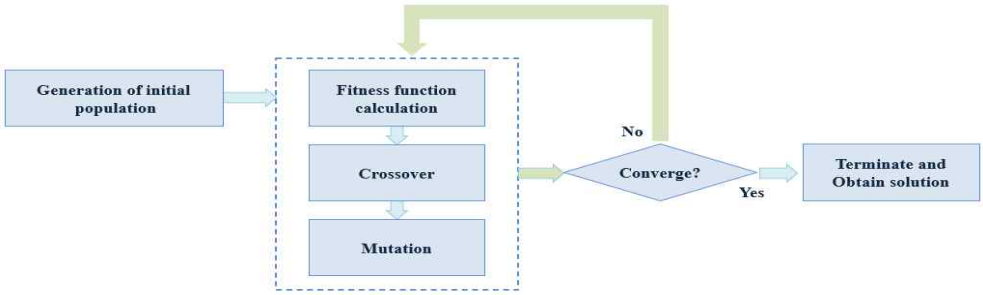

Fig. 2. 유전 알고리즘 도식도

유전 알고리즘의 구체적인 절차는 Table. 1.과 같다.[9] 유전 알고리즘의 선택 단계에서 무작위로 생성된 모집단은 다음 세대로 넘어갈 우월한 개체를 선택하는데 이때 적합도(fitness) 평가 과정을 거친다. 이후 교배 단계에서 부모라 불리는 두 개체의 유전자 일부를 교차하여 새로운 자손을 생성한다. 우월한 개체들의 유전자 교배 과정은 다음 세대에서 더 좋은 유전자를 갖는 개체를 생성하기 때문에 최적의 해를 찾는데 가장 중요한 과정이다.[10] 마지막 단계는 돌연변이 단계로, 개체가 가지고 있는 유전자를 확률적으로 변화시키는 단계이다. 돌연변이는 지역 최적화에 빠지는 상황을 방지하는 역할을 하는데, 선택과 교배과정으로부터 생성할 수 없는 새로운 개체를 만들고 잠재해의 다양성을 높인다. 일정 시간동안 해당 과정을 반복 수행하여 목적함수가 수렴하게 되면 최적화 문제의 해를 얻는다.

Table. 1. The procedure of a general GA[13]

Algorithm. Genetic Algorithm

Step 1 : Set $t = 1$. Randomly make N solutions to form the chromosome, P_1. Evaluate the fitness of solution in P_1.

Step 2 : ***Crossover***: Generate an offspring chromosomes Q_t as follows:

2.1 Select two solutions x and y from P_t based on the fitness values

2.2 Using a crossover operator, generate offspring and add them to Q_t

Step 3 : ***Mutation***: Mutate each solution $x \in Q_t$ with a predefined probability

Step 4 : ***Fitness assignment***: Evaluate and obtain a fitness value to each solution $x \in Q_t$ using its objective function value.

Step 5 : ***Selection***: Select N solutions from Q_t based on their fitness and copy them to P_{t+1}

Step 6 : If the object function is converged, terminate the search and return to the current best solution, else, set $t = t + 1$ go to *Step* 2 :

- 정책 그래디언트(Policy gradient)

정책 그래디언트(policy gradient)는 강화학습의 방법론 중 하나로, 마르코프 결정 프로세스(MDP, markov decision process)에서 파라미터화된 정책을 이용하여 누적 보상의 기댓값으로 이루어진 목적함수를 최대로 만드는 최적 정책(optimal policy)을 구하는 방법론이다.[11] MDP에서 에이전트와 환경의 상호작용 과정은 Fig. 3.과 같다.

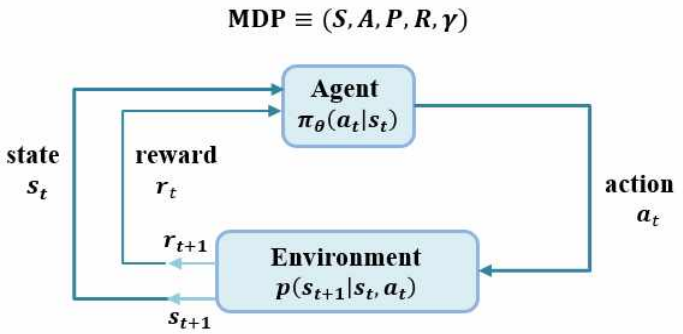

Fig. 3. The agent − environment interaction

본 연구에서 사용한 액터-크리틱(actor-critic) 방식의 알고리즘에서 MDP는 (S,A,P,R,γ) 네 가지 구성요소를 갖는다.[12] S는 상태(state), A는 행동(action)의 집합으로 연속공간 또는 이산공간 랜덤변수로 정의된다. R은 상태 s_t에서 행동 a_t를 선택했을 때 받는 보상(reward)함수이고 에피소드 종료 시점에 누적된 보상의 반환값(discounted return)은 식 (1)과 같다.

$$\begin{aligned} G_t &= r(s_t,a_t)+\gamma r(s_{t+1},a_{t+1})+\gamma^2 r(s_{t+2},a_{t+2})+\ldots+\gamma^{T-t} r(s_T,a_T) \\ &= \sum_{k=t}^{T}\gamma^{k-t} r(s_k,a_k) \end{aligned} \tag{1}$$

여기서, $r(s_t,a_t)$는 t시간 상태 s_t에서 행동 a_t를 수행할 때 주어지는 보상이며, $\gamma\in[0,1]$는 감가율(discount factor)로 미래의 가치에 대한 불확실성을 반영한다.

P는 상태 상태전이 확률(state transition probabilities)로, 상태 s_t에서 행동 a_t를 선택했을 때 다음 시간스텝의 상태 s_{t+1}로 갈 확률밀도함수이다. 상태와 행동이 이산공간 랜덤변수일 경우, 상태전이 확률밀도함수는 현재 상태와 행동의 선택으로 다음 상태가 될 확률로 표현된다.

$$P = p(s_{t+1}|s_t,a_t) = p(s_{t+1}|s_t,s_{t-1},..,s_0,a_t,a_{t-1},..,a_0) \tag{2}$$

상태전이 확률 P는 식 (2)와 같다. 위 식은 MDP에서 미래의 상태 s_{t+1}는 t시간 이전의 과거의 상태와 행동에 무관하며 오직 상태 s_t와 행동 a_t에만 영향을 받는다는 것을 나타낸다.

$$\pi(a_t|s_t) = p(a_t|s_t) \tag{3}$$

정책 π은 에이전트가 상태를 관측하여 취할 수 있는 행동을 결정짓는 조건부 확률 밀도함수로 식 (3)과 같다. 정책의 정의로부터 각 상태에서 에이전트가 선택할 수 있는 행동이 여러 개가 될 수 있다는 것을 알 수 있다. 이러한 경우를 확률적 정책이라 한다.[13]

$$\begin{aligned} V^{\pi}(s) &= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s] \\ &= E_{\pi}[\sum_{t}^{T} \gamma^{t-1} r(s_t, a_t) | S_t = s] \end{aligned} \tag{4}$$

여기서, S_t는 시간 t에서의 상태이다. 즉, 상태가치 함수는 t시간 이후부터 종료시점 까지 상태 S_t에서 정책 π로 얻어지는 총 보상의 기댓값이다.

가치함수는 상태가치(state-value) 함수 $V^{\pi}(s)$와 행동가치(action-value) 함수 $Q^{\pi}(s,a)$로 구분되는데, 강화학습 에이전트는 정책 π에 의해 행동이 수행되었을 때 얻는 누적 보상의 반환값의 기댓값인 가치함수가 최대가 되는 것을 목표로 한다.[14]

행동가치 함수 $Q^{\pi}(s,a)$는 식 (5)와 같이 정의된다.

$$\begin{aligned} Q^{\pi}(s,a) &= E_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | S_t = s, A_t = a] \\ &= r(s_t, a_t) + E_{s_{t+1}}[V(s_{t+1})] \end{aligned} \tag{5}$$

여기서, S_t, A_t는 시간 t에서의 상태와 행동이다. 행동가치 함수는 t시간 이후부터 종료시점 까지 상태 S_t에서 행동 A_t를 수행한 후 정책 π로 얻어지는 총 보상의 기댓값이다.

최적의 정책 π^*을 구하는 것은 목적함수 $J(\theta)$를 최대로 만드는 정책 파라미터 θ를 계산하는 것과 같다. 파라미터화 된 정책 $\pi_{\theta}(a_t, s_t)$을 갖는 MDP의 전체 반환값을 목적함수로 갖는 $J(\theta)$는 식 (6)과 같다.[17]

$$J(\theta) = E_{s_t, a_t}[\log_{\pi_\theta}(a_t|s_t) \cdot A(s_t, a_t)] \quad (6)$$

여기서, $A(s_t, a_t)$는 시간 t에서 선택한 행동이 반환값의 기댓값에 미치는 영향의 척도이며 $A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$로 정의되는 이점 함수(advantage function)이다.

식 (7)는 목적함수 $J(\theta)$를 정책 파라미터 θ에 대하여 미분한 것이다.

$$\nabla_\theta J(\theta) = E_{s_t, a_t}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)(\sum_{k=t}^{T} \gamma^{k-t} r(s_t, a_t))] \quad (7)$$

$\nabla_\theta J(\theta)$는 목적함수의 그래디언트이며, 이를 최대로 하는 파라미터 θ는 식 (8)로 정의된 경사상승법(gradient ascent)을 이용하여 계산한다.

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \quad (8)$$

이와 같이 정책 그래디언트란 정책 파라미터 θ로 정의된 목적함수를 최대화 하여 최적의 정책 π^*을 구하는 과정이다. 강화학습에 정책 그래디언트를 이용할 경우, 분산이 감소하므로 안정적인 학습이 가능하다.[15]

Offset Point Interpolation using Reinforcement Learning

Yeonjin Jang
School of Naval architecture and Ocean engineering, University of Ulsan

ABSTRACT

B-spline curve interpolation is widely used in various fields such as computer graphics, computer-aided design, and robotics. The shape of the interpolated curve heavily relies on the control points calculated based on the parameterization method. When determining the parameter values(knots) that affect the shape of the interpolated curve in B-spline curve interpolation, different parameterization methods lead to distinct calculations of control points. This characteristic makes it challenging to precisely represent the desired curve shape. In this study, to overcome these limitations, proposes a novel parameter optimization method based on reinforcement learning when performing B-spline curve interpolation. The proposed method generates section curves based on the calculated parameter values applied to offset data, representing hull form. The agent in the reinforcement learning adjusts normalized parameters iteratively from 0 to 1 and defines transformed parameter values as states. Observing state changes, the agent learns the optimal parameter values for a given point by observing environment feedback, represented by the resulting curve. The agent refines the optimal parameter values for the given B-spline through iterative adjustments of parameter values and observing resulting curves. To evaluate the proposed method, we compared the results obtained using the conventional optimization method, genetic algorithm, model with those obtained applying the proposed algorithm. Furthermore, we compared the proposed method with existing parameterization methods when generating section curves based on offset data to validate the usefulness of the proposed approach. The proposed parameterization optimization method demonstrated superior performance compared to existing methods. This approach presented in this study offers a novel approach to finding optimal parameters when interpolating B-spline curves, showing the potential of utilizing machine learning techniques in geometric modeling.