한국지능시스템학회 논문지 2011, Vol. 21, No. 6, pp. 737-742
http://dx.doi.org/10.5391/JKIIS.2011.21.6.737

주행문제를 위한 최신 근사적 동적계획법의 적용

Application of Recent Approximate Dynamic Programming Methods for Navigation Problems

민대홍*, 정근우*, 권기영*, 박주영*
Daehong Min, Keun-woo Jung, Ki-young Kwon, Jooyoung Park

* 고려대학교 제어계측공학과

요 약

주행문제는 불확실한 외란의 영향을 받는 이동로봇 등의 시스템에 대하여 각종 제약조건들을 만족하도록 하는 제어 입력을 결정하는 과제를 포함한다. 이러한 과제는 제약조건을 갖는 확률 제어 문제로 모델링될 수 있다. 이러한 종류의 제어 문제를 풀기 위하여, 최적 가치함수 개념에 의존하는 동적 계획법의 활용을 시도해볼 수 있다. 하지만, 대부분의 실제 문제에서 이러한 시도는 많은 어려움, 예컨대, 시스템의 완벽한 모델이 알려지지 않거나, 최적 제어정책을 구하기 위한 계산이 불가능하거나, 수없이 많은 계산 자원을 필요로 하는 등의 문제점을 안겨줄 수 있다. 이러한 동적 계획법의 어려움을 극복하기 위한 전략으로, 근사적 가치함수에 의존하여 준최적 제어정책을 구하는 근사적 동적 계획법을 사용할 수 있다. 본 논문에서는 최근에 제시된 근사 동적 계획법들을 복잡한 제약조건을 갖는 주행 문제에 적용하여 보고 그 결과로 얻어지는 성능 특성을 관찰해 본다.

키워드 : 근사적 동적계획법, 모델예측제어, 확률제어, 주행

Abstract

Navigation problems include the task of determining the control input under various constraints for systems such as mobile robots subject to uncertain disturbance. Such tasks can be modeled as constrained stochastic control problems. In order to solve these control problems, one may try to utilize the dynamic programming(DP) methods which rely on the concept of optimal value function. However, in most real-world problems, this trial would give us many difficulties; for examples, the exact system model may not be known; the computation of the optimal control policy may be impossible; and/or a huge amount of computing resource may be in need. As a strategy to overcome the difficulties of DP, one can utilize ADP(approximate dynamic programming) methods, which find suboptimal control policies resorting to approximate value functions. In this paper, we apply recently proposed ADP methods to a class of navigation problems having complex constraints, and observe the resultant performance characteristics.

Key Words : Approximate dynamic programming, Model predictive control, Stochastic control, Navigation

1. 서 론

동적 계획법을 이용하여 확률 시스템(stochastic system)의 최적 제어 정책을 얻기 위해서는 최적 가치함수(optimal value function)를 구하는 문제를 다뤄야 한다. 하지만 이러한 최적 가치함수를 얻기 위해서는 완벽한 시스템 모델을 바탕으로 가치함수를 설계해야 하고, 완벽한 시스템 모델을 알고 있다 하더라도 최적 가치함수를 계산하기 위해서는 가능한 모든 상태에 대해서 고려해야 하므로 계산이 불가능 하거나 지나치게 많은 계산 자원을 필요로 하는 문제가 생길 수 있다[1].

접수일자 : 2011년 11월 19일
완료일자 : 2011년 12월 11일
본 논문은 본 학회 2011년도 추계학술대회에서 선정된 우수논문입니다.
감사의 글 : 본 연구는 지식경제부 융복합형 로봇인력 양성 "로봇자율주행기술연구센터" 지원사업의 연구결과로 수행되었음
(NIPA-2011-C7000-1001-0005)

최근에 이러한 동적 계획법의 현실적인 한계를 극복하기 위한 방법 중 하나로, 가치함수를 근사화된 가치함수로 대체하여 확률 제어 문제(stochastic control problem)의 준 최적 제어 정책(suboptimal control policy)을 구하는 근사 동적 계획법(approximate dynamic programming, ADP)이 소개된 바 있다[1][2][3].

주행 문제(navigation problem)는 로봇 분야의 주요한 연구주제 중 하나로, 다양한 방향(e.g., [4][5])의 연구가 진행되고 있다. 본 논문에서는 현재 위치에 대하여 불확실성(uncertainty)을 갖는 차량 혹은 이동로봇을 목표 위치로 이동하도록 하는 제어 입력을 얻어내는 문제를 고려한다. 이러한 주행 문제에서는 여러 가지 제약조건을 고려해야 하는 부분이 특히 중요하다. 이러한 제약조건의 예로는 장애물과 같은 위치 제약조건과 시스템의 특성에서 발생할 수 있는 입력 제약 조건 등을 들 수 있다. 본 논문에서는 최근에 모델 예측 제어(model predictive control, MPC)를 기반으로 근사 동적 계획법[3]을 적용한 사례가 있는 복합적인 제약조건을 갖는 주행문제[6]의 결과를 재현하고, 해당 문제를 복합적인 제약조건을 갖는 확률 제어 문제로 접근하여 최근에 소개된 Min-max ADP[2]를 적용한다. 그리고 방법론들의 결

과를 비교 및 관찰한다.

2. 주행문제 모델의 정의

본 논문에서는 [6]에서 다루고 있는 주행문제 모델을 고려한다.

$$x_{k+1}=f(x_k,u_k,w_k)=Ax_k+Bu_k+w_k,\quad k=0,1,2,\cdots \tag{1}$$

여기에서 $x_k\in\mathcal{R}^n$는 위치이고 $u_k\in\mathcal{R}^m$는 제어 입력이며, $w_k\in\mathcal{R}^n$는 독립동일분포(independent and identically distribution)를 갖고 다중 정규분포를 따르는 외란(disturbance)이다.

만약 이산시간의 단위 시간간격 Δ 에 대해 고려한다고 하면, 다음과 같이 시스템 행렬과 외란을 정의 할 수 있다[6].

$$\begin{aligned} A\in\mathcal{R}^{n\times n},\quad & B\in\mathcal{R}^{n\times m},\\ w_k\sim N(0,\Delta\Omega^w),\quad & \Omega^w\in\mathcal{R}^{n\times n}>0 \end{aligned} \tag{2}$$

센서 모델은 다음과 같이 표현하며, 측정 가능한 특징점(feature)으로부터 시스템까지의 거리를 제공한다.

$$z_k=h(x_k)+v_k,\quad k=0,1,2,\cdots \tag{3}$$

그리고 이동로봇이 갖는 위치 x_k 와 입력 u_k 에 대한 제약조건은 각각 식(4)와 같이 표현할 수 있다.

$$\begin{aligned} x_{\min}\le x_k\le x_{\max}\\ u_{\min}\le u_k\le u_{\max} \end{aligned} \tag{4}$$

3. MPC ADP를 이용한 제어기 설계

MPC ADP를 이용한 제어기는 유한 구간(finite horizon) T 에 대해서, 식(5)와 같이 설계할 수 있다[6].

$$\begin{aligned} \text{minimize}\quad & E\left\{g_d(x_{k+T})+\sum_{i=k}^{T}g_i(x_i,u_i,\overline{w})\middle|I_k\right\},\\ \text{subject to}\quad & x_{k+1}=f(x_k,u_k,\overline{w}),\\ & x_{\min}\le x_{i+1}\le x_{\max},\\ & u_{\min}\le u_i\le u_{\max}, \end{aligned} \tag{5}$$

이와 같이 설계한 최적화 문제 식(5)의 목적함수는 짧은 유한 구간에 대하여 가치함수를 최적화하는 제한된 룩어헤드 방법론(limited lookahead method)을 도입하여, 현재 상태와 제어 입력에 대한 단계 비용함수(stage cost) $g_i(x_i,u_i,\overline{w})$와 유한 구간(finite horizon) 이후의 종말 비용함수(terminal cost) $g_d(x_{k+T})$를 이용해서 구성한다. 그리고 이러한 제한된 룩어헤드 방법론에 확실성 등가 제어(certainty equivalent control)의 개념을 도입하여 계산을 좀 더 효과적으로 수행할 수 있도록 현재 시간 k에 대한 외란의 기댓값 $\overline{w}$를 적용한다[3].

또한, 본 논문에서 고려하는 시스템은 확률 시스템으로 현재 상태에 대한 정확한 정보를 알 수 없다. 그러므로 외부 관측 정보를 도입하여, 관측 정보와 이전 시간의 제어 정책을 기반으로 구성된 정보 벡터(information vector) I_k 를 도입하여 오직 현재 상태에 대한 조건부 확률 $P_{x_k|I_k}$ 만을 구하고 이것을 바탕으로 현재 상태를 정의한다. 그리고 목적함수를 최적화 하는 과정에 있어서 요구되는 미래의 관측 정보는 무시하고, 시스템 모델을 이용한 개루프 최적화(open-loop optimization) 방법론을 사용하는 개루프 되먹임 제어(open-loop feedback control)방법론을 고려한다[3].

마지막으로 식(5)와 같은 최적화 과정을 통하여 제어 정책을 얻어내기 위해서는 시스템이 임의의 시간 m이 경과한 이후에 목표 영역에 도달하도록 하는 제어 입력이 존재해야 하고, 목표 영역에 도달할수록 0에 근접한 제어 입력을 제공해야 한다. 그리고 정의된 목적함수가 무한 구간(infinite horizon)에 대해서 유한한 합을 갖는다고 가정한다. 이러한 가정을 만족하는 상황에 대해서 식(5)를 최적화 하는 제어 정책을 모델 예측 제어 정책(model predictive control policy)라고 부른다[3]. 그리고 이렇게 구한 제어 정책 중 현재 시간 k에 대한 제어 입력만을 취하는 전략을 사용한다.

4. Min-max ADP를 이용한 제어기 설계

Min-max ADP 방법론은 식(6)과 같은 최적화 문제로부터 근사 가치함수 $\hat{V}$를 구한다[2].

$$\begin{aligned} \text{maximize}\quad & \inf_u\left(\ell(x,u)+\gamma\underset{w}{E}\hat{V}(f(x,u,w))\right)\\ \text{subject to}\quad & \hat{V}\le V^\star \end{aligned} \tag{6}$$

여기에서 $\ell(x,u)$는 비용 함수이고, γ는 할인율(discount rate)이다. 그러나 식(6)을 풀기 위해서는 최적 가치함수 $V^\star$를 알아내야 하는 문제가 생긴다. 이러한 문제를 해결하기 위한 방법으로 아래 식(7)과 같은 벨만 연산자(Bellman operator)의 성질을 고려한다.

$$(Th)(z)=\inf_{u\in U}\{\ell(x,u)+\gamma Eh(f(x,u,w))\} \tag{7}$$

이러한 벨만 연산자는 어떠한 함수 $h:x\to\mathcal{R}$에 대해서도 성립하며 단조성(monotonicity)과 연산자의 반복 적용에 의한 고정점으로 수렴하는 성질을 갖는다[1]. 즉, 벨만 연산자를 가치 함수 V에 무한히 적용하면, 어느 한 점으로 수렴하게 되고, 최종적으로 수렴하게 되는 가치함수를 최적 가치함수 $V^\star$로 정할 수 있다. 그러나 벨만 연산자를 무한히 적용하여 수렴하는 가치함수를 얻어내는 방법은 현실적으로 많은 어려움이 따르게 되므로 다음 식(8)과 같은 후보 가치함수에 벨만 연산자를 적용하는 방법을 고려한다[1][2].

$$\hat{V}_\alpha=\sum_{i=1}^{K}\alpha_i V^{(i)} \tag{8}$$

여기에서 $V^{(i)}$는 기저(basis) 함수 이다.

후보 가치함수에 벨만 연산자 T의 성질을 이용하면 다음과 같은 관계식을 얻어낼 수 있다.

$$\hat{V}_\alpha \le T\hat{V}_\alpha \tag{9}$$

이러한 부등식을 벨만 부등식 이라고 한다. 그리고 이러한 벨만 부등식을 정수 $M \ge 1$ 만큼 반복해서 적용한 형태를 반복적 벨만 부등식(iterated Bellman inequalities)이라고 하고, 식 (10)과 같이 표현할 수 있다[1][2].

$$\hat{V}_\alpha \le T^M \hat{V}_\alpha \tag{10}$$

그리고 반복적 벨만 부등식은 식 (11)과 같은 성질에 의해서 조건 $\hat{V}_\alpha \le V^\star$를 만족하는 충분조건을 제공해 준다[1][2].

$$\hat{V}_\alpha(x_0) \le \lim_{k\to\infty}(T^{kM}\hat{V}_\alpha)(x_0) = V^\star(x_0),\ \ \forall x_0 \in X \tag{11}$$

주의할 점은 여기에서 k는 이산 시간에 대한 것이 아닌 벨만 연산자를 적용한 회수를 의미한다. 그리고 여기에서 x_0는 제약조건 X에 존재하는 현재 상태를 의미한다.

반복적 벨만 부등식을 이용하여 근사 가치함수를 구하기 위한 최적화 문제 식 (6)을 표현하면 다음과 같다[2].

$$\begin{array}{ll} \text{maximize} & \inf_u \left(\ell(x,u) + \gamma \mathop{E}_w \hat{V}(f(x,u,w))\right) \\ \text{subject to} & \hat{V}_{i-1} \le T\hat{V}_i,\ \hat{V}_{M-1} = \hat{V}_M,\ i = 1,\cdots M \end{array} \tag{12}$$

그리고 식 (12)의 목적함수를 최적화 하는 가치함수 $\hat{V}$은 최적 가치함수의 최대 하한이라고 할 수 있으므로, 근사 가치함수로 사용할 수 있다.

본 논문에서 구하고자 하는 것은 비용함수로 구성되는 최적 가치함수를 최소화 하는 제어 정책 이다. 최적화 문제 식 (12)를 통해서 최적 가치함수에 근접한 근사 가치함수 $\hat{V}$을 최소화 하는 제어 정책을 제약조건 내에서 구하는 문제로 이러한 목적을 달성할 수 있으며, 이러한 최적화 문제는 아래 식 (13)과 같이 표현할 수 있다.

$$\begin{array}{ll} \text{minimum} & \ell(x_k,u_k) + \gamma E\hat{V}(f(x_k,u_k,w_k)) \\ \text{subject to} & x_{k+1} = f(x_k,u_k,\overline{w}), \\ & x_{\min} \le x_{k+1} \le x_{\max}, \\ & u_{\min} \le u_k \le u_{\max} \end{array} \tag{13}$$

여기에서 k는 이산 시간의 현재 시점을 의미하며, $\overline{w}$는 MPC ADP에서와 같이 외란의 평균을 의미한다. 그리고 이렇게 구한 제어 정책 중 현재 시간 k에 대한 제어 입력 u_k만을 취하는 전략을 도입한다.

5. 시뮬레이션 및 결과

본 논문에서 고려하는 환경은 [6]에서 고려한 환경과 유사하다. 지도 정보를 알고 있으며, 경로 계획이 완료되어 이동로봇은 경로점(waypoint) 사이를 이동한다. 이때 확장 칼만 필터(extended Kalman filter)를 이용한 상태 추정치에 의존하여 본 논문에서 고려하는 ADP 방법론들로부터 얻어낸 제어 정책을 적용하는 방법으로 제어된다. 현재 상태를 추정하기 위한 측정치로는 임의의 특징점들(feature points)과 현재 차량의 거리를 이용하며, 특징점을 측정할 수 있는 구간 역시 제한되어 있다. 차량의 위치에 대한 제약조건은 경로 구간마다 변하며, 제어 입력의 제약 조건은 전 구간에서 $-1 \le u_k \le 1$ 이다. 이밖에 상세한 수치적 값들은 [6]을 따른다. 그리고, 본 논문에서 고려하는 각종 최적화 문제들의 해는 MATLAB을 기반으로 하여 얻어졌다.

5.1 MPC ADP 방법론의 적용

앞서 2장에서 정의한 주행문제를 MPC ADP 방법론으로 제어하기 위한 비용함수를 다음과 같이 정의한다[6].

$$\begin{aligned} g_d(x_N) &= (x_N - x_d)^T Q_d (x_N - x_d), \\ g(x_k,u_k,w_k) &= u_k^T R u_k \end{aligned} \tag{14}$$

여기에서 Q_d는 양의 정부호 행렬(positive definite matrix)이며, R은 양의 준정부호 행렬(positive semi definite matrix)이다. 그리고 N은 유한 구간의 크기를 나타내며, x_d는 현재 목표로 하는 경로점의 위치이다.

식 (14)와 같이 정의한 비용함수를 앞서 설계한 MPC ADP를 이용한 최적화 문제 식 (5)에 적용하여 시간 k에서의 상태 x_k에서 목적함수를 최소화 하는 제어 입력 u_k를 적용한다.

5.2 Min-max ADP 방법론의 적용

주행문제를 위한 Min-max ADP 제어기의 비용함수를 다음 식 (15)과 같이 정의한다.

$$\ell(x_k,u_k) = (x_k - x_d)^T Q_d (x_k - x_d) + u_k^T R u_k \tag{15}$$

여기에서 Q_d와 R은 MPC ADP의 경우와 동일하다.

그리고 최적 가치함수의 최대 하한을 구하기 위한 후보 가치함수를 다음과 같은 이차식 형태로 정의한다[1][2].

$$\hat{V}(x_k) = x_k^T P x_k + 2p^T x_k + s \tag{16}$$

이와 같은 내용을 식 (12)에 적용하여 최적 가치함수의 최대하한을 만족하는 근사 가치함수 $\hat{V}(x_k)$를 구하고, 식 (13)에 적용하여 시간 k에서의 상태 x_k에 대해서 근사 가치함수를 최소화 하는 제어 정책 u_k를 구한다.

5.3 시뮬레이션 결과

그림 1은 [6]의 주행문제 환경에 대해 MPC ADP 제어기를 적용한 이동로봇의 이동 궤적을 나타내고 있다. 그리고 그림 2는 이러한 이동 궤적을 만들어내는 MPC ADP 제어기의 제어 정책이다.

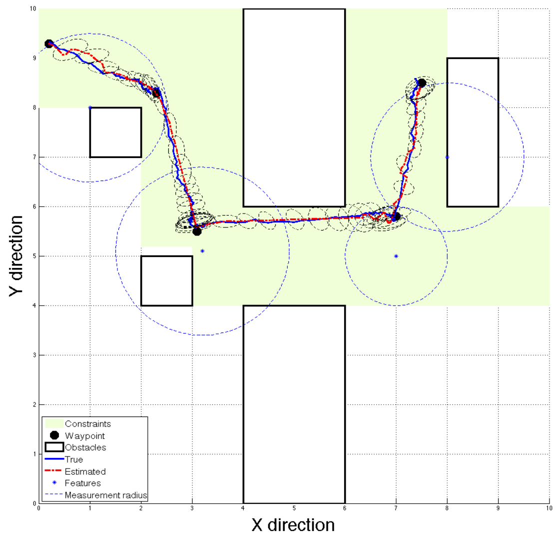

그림 1. MPC ADP 방법론을 이용한 주행궤적 (주행 환경 1[6])

Fig. 1. Navigation trajectory using MPC ADP method (navigation environment 1[6])

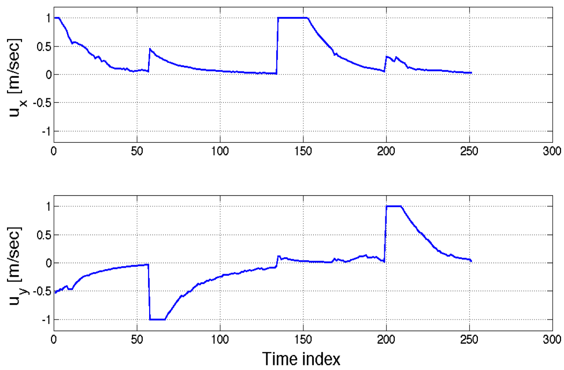

그림 2. MPC ADP 방법론의 제어 정책 (주행 환경 1[6])

Fig. 2. Control policy of MPC ADP method (navigation environment 1[6])

그림 1과 2는 [6]의 결과를 재현한 것이다. MPC ADP 방법론이 제약조건들을 만족하는 제어 정책을 구사하며 적절하게 궤적을 제어 하는 것을 관찰할 수 있다.

그림 3은 같은 주행 환경에 대해서 Min-max ADP 방법론으로 제어한 주행문제의 이동 궤적을 나타낸다. 그리고 그림 4는 이러한 환경에서의 Min-max ADP 방법론의 제어 정책이다.

Min-max ADP 제어기를 적용한 입력 제약조건을 만족하는 제어 입력을 이용하여, 위치를 제약조건 안에서 제어하며 목표 지점에 도달하는 것을 확인할 수 있었다. 그림 2와 그림 4에서 관찰 할 수 있듯이 Min-max ADP의 경우가 MPC ADP 보다 목표지점에 더 빠르게 도달하도록 하는 제어 정책을 구사하는 것을 관찰할 수 있다.

이와 같이 경로점을 직선으로 연결이 가능한 경우에 각 ADP 방법론 들은 성공적으로 시스템을 제어하는 것을 관찰할 수 있었다.

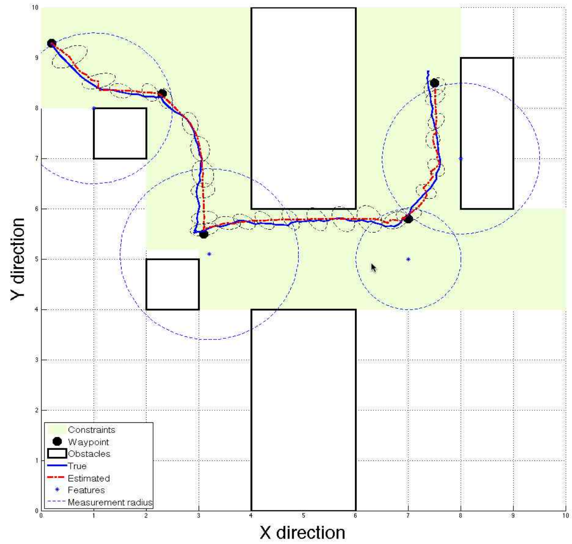

그림 3. Min-max ADP 방법론을 이용한 주행궤적 (주행 환경 1[6])

Fig. 3. Navigation trajectory using min-max ADP method (navigation environment 1[6])

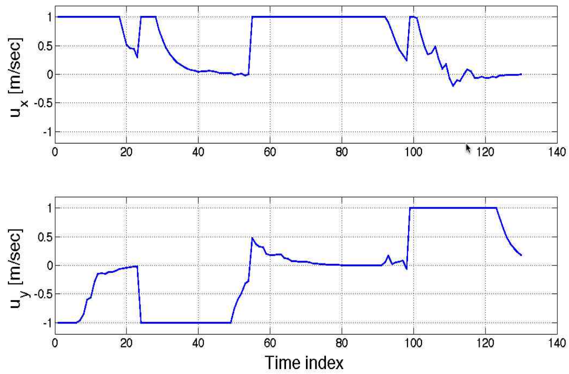

그림 4. min-max ADP 방법론의 제어 전략 (주행환경 1[6])

Fig. 4. Control policy of min-max ADP method (navigation environment 1[6])

그림 5는 경로점 사이를 위치 제약조건의 영향에 의해서 직선으로 이동할 수 없는 환경에 대해 MPC ADP를 적용한 결과이다. 또한, 그림 6은 그림 5에 대한 MPC ADP 방법론의 제어 전략이다.

그림 5에서 제약조건이 이동구간 사이를 가로 막는 경우 이러한 제약조건을 미리 고려하는 제어 입력을 적용하는 모습을 확인할 수 있다. 즉, MPC ADP 방법론의 특징 중 하나인 유한구간에 대해서 미리 시스템 모델을 이용하여 진행하여 미래의 상태가 현재의 제어 입력에 영향을 주도록 하는 이론적 특성이 이와 같은 결과 그림에서 잘 나타나는 것을 관찰할 수 있다.

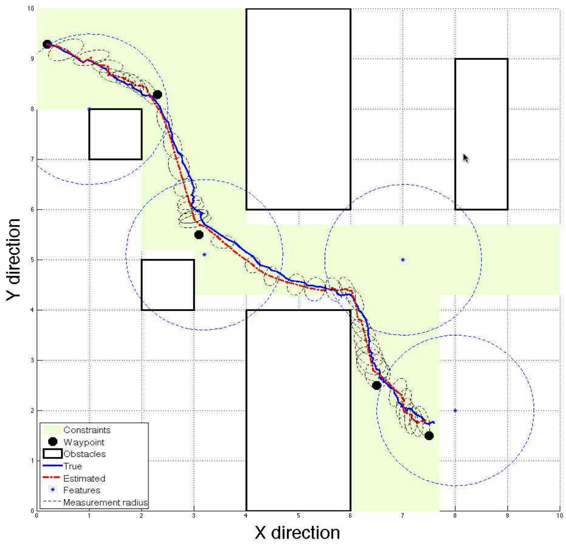

그림 5. MPC ADP 방법론을 이용한 주행궤적 (주행 환경 2)

Fig. 5. Navigation trajectory using MPC ADP method (navigation environment 2)

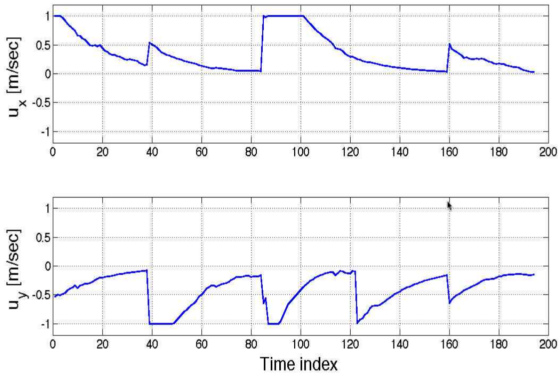

그림 6. MPC ADP 방법론의 제어 전략 (주행 환경 2)

Fig. 6. Control policy of MPC ADP method (navigation environment 2)

반면에 Min-max ADP의 경우 제약 조건의 경계를 따라서 이동하는 제어 정책을 얻어내는 것을 그림 7에서 관찰할 수 있다. 그리고 이러한 궤적에 대한 제어 정책을 그림 8에서 관찰할 수 있으며, 앞서 관찰한 주행 환경 1 에서와 같이 MPC ADP 방법론의 제어 정책 보다 더 빠르게 목표지점으로 도달하도록 하는 제어 정책을 얻어내는 모습을 관찰할 수 있다.

6. 결론 및 향후 연구 과제

본 논문에서는 최근에 소개된 ADP 방법론들을 설명하고, 복합적인 제약조건이 존재하는 주행문제를 풀어보았다. 그리고 각 각의 ADP 제어기 들을 이용하여 이러한 문제에 대해서 효과적인 제어 정책을 결정할 수 있는 것으로 판단할 수 있었다.

MPC ADP의 경우 미래의 제약조건을 현재의 제어 정책에 미리 반영하는 모습을 관찰하였고, Min-max ADP의 경우 제약조건의 경계를 따라서 움직이는 제어 정책을 보이는 경향이 관찰되었으며, MPC ADP 보다 좀 더 빠르게 목표 지점에 도달하는 제어 정책을 얻어내는 것을 관찰하였다.

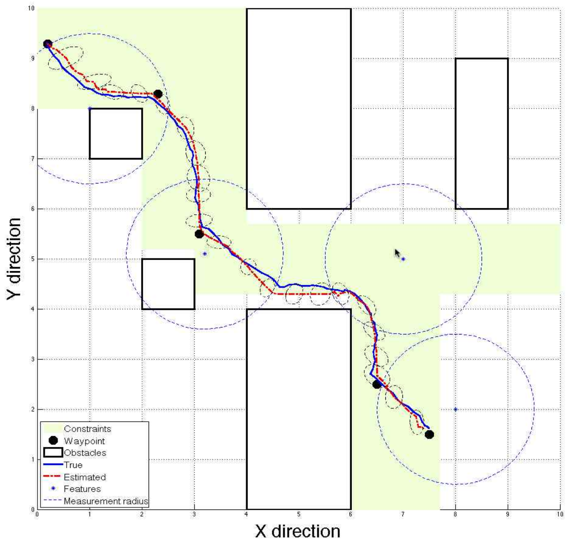

그림 7. Min-max ADP 방법론을 이용한 주행궤적 (주행 궤적 2)

Fig. 7. Navigation trajectory using min-max ADP method (navigation environment 2)

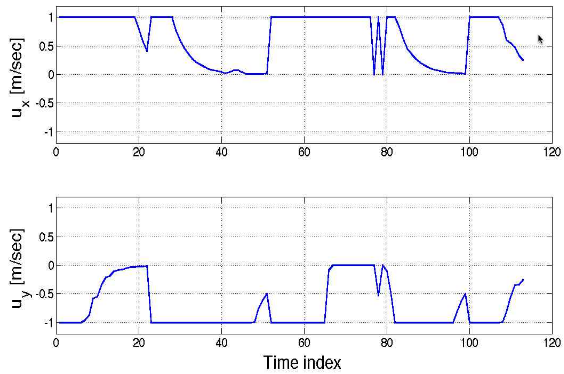

그림 8. Min-max ADP 방법론을 이용한 주행궤적 (주행 환경 2)

Fig. 8. Navigation trajectory using min-max ADP method (navigation environment 2)

하지만, 관찰한 ADP 방법론들 모두, 위치 추정값이 실제 위치보다 오차가 일정 범위 이상으로 큰 경우에는 실제 시스템이 제약조건을 벗어나는 경우가 생기는 것을 관찰하였다.

향후 좀 더 구체적인 이동로봇의 모델 또는 실제 이동로봇에 이러한 최신 기법의 ADP 방법론들을 적용하면 더욱 객관적이고 실질적인 방법론의 평가가 가능할 것이다. 더 나아가 이러한 ADP 방법론들의 장점들을 결합한 새로운 ADP 방법론들에 대한 연구도 가능할 것으로 기대된다.

참 고 문 헌

[1] Y. Wang and S. Boyd, "Approximate dynamic programming via iterated Bellman inequalities," *Stanford University, 2010. (Available at http://www.stanford.edu/~boyd/papers.html)*

[2] B. O'Donoghue, Y. Wang, and S. Boyd, "Min-max approximate dynamic programming," *Proceedings of 2011 IEEE International Symposium on Computer-Aided Control System Design,* pp. 424-431, 2011.

[3] D. P. Bertsekas, "Dynamic programming and suboptimal control: A survey from ADP to MPC," *European Journal of Control, vol. 11, nos. 4-5,* pp. 310-334, 2005.

[4] 김정민, 허정민, 정승영, 김성신, "자율주행 장치를 위한 수정된 유전자 알고리즘을 이용한 경로계획과 특징 맵 기반 SLAM," 한국지능시스템학회 논문지, 19권 3호, pp. 381-387, 2009.

[5] 나두영, 노수희, 문형필, 정진우, 김용태, "모듈형 로봇의 자가 결합을 위한 퍼지 주행 제어 및 장애물 회피 제어", *한국지능시스템학회 논문지,* 19권 4호, pp. 470-477, 2009.

[6] V. A. Huynh, "Navigation in GPS-denied environments using approximate dynamic programming," *MIT, 2009. (Avilable at http://www.mit.edu/~vuhuynh/publications.html)*

저 자 소 개

민 대 홍 (Daehong Min)
2010년 고려대학교 제어계측공학과 (공학사)
2011년~현재 고려대학교 제어계측공학과 석사과정

관심분야 : 지능 제어, 기계 학습, 제어 이론
E-mail : min2003@korea.ac.kr

정 근 우 (Keun-woo Jung)
2010년 고려대학교 제어계측공학과 (공
2011년~현재 고려대학교 제어계측공 석사과정
관심분야 : 지능 제어, 기계 학습

E-mail : rmsdn@korea.ac.kr

권 기 영 (Ki-young Kwon)
2011년 고려대학교 제어계측공학과 (공학사)
2011년~현재 고려대학교 제어계측공학과 석사과정

관심분야 : 강화학습, 모델예측제어
E-mail : kinnkin@korea.ac.kr

박 주 영 (Jooyoung Park)
1983년 서울대학교 전기공학과 (공학사)
1985년 KAIST 핵공학과 (공학석사)
1992년 University of Texas at Austin 전기및컴퓨터공학과 (공학박사)
1993년~현재 고려대학교 제어계측공학과 교수

관심분야 : 기계학습, 제어 및 시스템이론
E-mail : parkj@korea.ac.kr