2011년도 대한전기학회 하계학술대회 논문집 2011. 7. 20 - 22

이산 시간 스위칭 선형 시스템의 적응 LQ 준최적 제어를 위한 Q-학습법

전태윤*, 최윤호**, 박진배*
연세대학교 전기전자공학과*, 경기대학교 전자공학과**

Q-learning for Adaptive LQ Suboptimal Control of Discrete-time Switched Linear System

Tae Yoon Chun*, Yoon Ho Choi**, Jin Bae Park*
Dept. of Electrical and Electronic Eng., Yonsei University*, Dept. of Electronic Eng., Kyonggi University**

Abstract - 본 논문에서는 스위칭 선형 시스템의 적응 LQ 준최적 제어를 위한 Q-학습법 알고리즘을 제안한다. 제안된 제어 알고리즘은 안정성이 증명된 기존 Q-학습법에 기반하며 스위칭 시스템 모델의 변수를 모르는 상황에서도 준최적 제어가 가능하다. 이 알고리즘을 기반으로 기존에 스위칭 시스템에서 고려하지 않았던 각 시스템의 불확실성 및 최적 적응 제어 문제를 해결하고 컴퓨터 모의실험을 통해 제안한 알고리즘의 성능과 결과를 검증한다.

1. 서 론

근사 동적 계획법(Approximate Dynamic Programming: ADP)은 최적 제어 문제의 해결을 위한 방법 중 하나로, 다양한 연구 결과에 의해서 여러 제어 분야에서 효과적임이 확인되었다 [1]. 여러 ADP 알고리즘 중 Q-학습법(Q-learning)으로 알려진 행동 종속적 휴리스틱 동적 계획법(action-dependent heuristic dynamic programming)은 동적 계획법(dynamic programming)이 가진 'curse of dimensionality' 문제를 완화시켜줄 뿐만 아니라 모델의 정보를 모르는 경우에도 반복적인 연산을 통해 최적 제어 문제를 해결할 수 있는 방법이다. 최근에 와서는 기존에 다루지 못했던 안정성 문제를 고려한 근사 동적 계획법 및 Q-학습법이 개발되었다 [2].

한편, 스위칭 시스템(switched system)은 스위칭 신호에 의해 변화하는 유한 개의 하부시스템(subsystem)로 이루어진 동적 시스템으로, 전력 전자, 임베디드 시스템, 통신 네트워크 분야와 같은 다양한 공학 분야에서 많은 관심을 받고 있다 [3]. 이러한 스위칭 시스템에서는 스위칭 변화에 따른 여러 개의 동적 모델이 존재하므로, 각각의 맞는 모델링 및 제어기 설계가 필요하다. 하지만 효과적인 모델링 과정은 때로는 매우 복잡하여 많은 시간을 요구하고, 그 결과는 불확실성 및 오차가 항상 존재한다. 따라서 이를 보완할 수 있는 효과적인 제어 방법이 요구되고 있다.

본 논문에서는 Q-학습법을 이용한 스위칭 선형 시스템의 적응 LQ 준최적 제어 알고리즘을 제안한다. 제안된 제어 알고리즘을 통해 스위칭 시스템의 여러 하부시스템을 모르는 경우에도 정의한 비용함수를 최소화시키는 관점에서의 준최적 제어 성능을 이끌어낼 수 있다. 이러한 결과를 모의실험을 통해 실제 수식 값과 그래프로 나타내고 이를 바탕으로 제안한 제어기의 성능과 효용성을 검증하고자 한다.

2. 본 론

2.1 이산 시간 스위칭 선형 시스템 모델

본 절에서는 일반적인 이산 시간 스위칭 선형 시스템(discrete-time switched system) 모델을 제시한다. 이산 시간 스위칭 선형 시스템은 다음과 같이 여러 개의 하부시스템으로 표현된다 [3].

$$x_{k+1} = A_j x_k + B_j u_k,\ k \in Z^+, j \in L = \{1,\ldots,N\} \quad (1)$$

여기서, 상태 변수 $x \in R^n$, 입력 $u \in R^m$ 이고, Z^+은 음수가 아닌 정수, 그리고 L은 유한한 인덱스의 집합으로, 각 모델들의 집합을 의미한다. 하부시스템 사이에 스위칭 신호는 임의의 조각 상수 맵 $\sigma_k \in Z^+ \rightarrow L$으로 나타낼 수 있으며, σ_k 신호에 따라 A_j, B_j가 변화하게 된다.

2.2 제안한 제어 알고리즘

본 절에서는 앞에서 설명한 이산 시간 스위칭 선형 시스템 모델에 대한 Q-학습법을 유도한다. Q-학습법을 스위칭 모델의 하부시스템인 (A_j, B_j)에 대해 각각 유도함으로써 각 하부시스템의 최적의 제어기를 설계한다. 그리고 이를 바탕으로 전체 시스템의 준최적 제어 기법을 유도한다.

우선, (1)에 대한 j번째 하부시스템 $(A_j, B_j, j \in L)$에 대한 비용함수(cost function)를 다음과 같이 정의하자.

$$V_j(x_k) := \sum_{i=k}^{\infty} [x_k^T S_j x_k + u_{j,k}^T R_j u_{j,k}]$$

여기서 S_j는 $n \times n$인 준양한정(positive semidefinite) 행렬이며, R_j는 $m \times m$인 양한정(positive definite) 행렬이다. j번째 하부시스템의 비용함수 $V_j(x_k)$를 최소화하는 입력 $u_{j,k}$를 $u_{j,k}^*$로 정의하면 최적의 비용함수 값은 $V_j^*(x_k) := \min_{u_j} V_j(x_k)$ 와 같이 정의할 수 있다. Bellman의 최적의 원리(optimality principle)에 의해, $V_j^*(x_k)$는 아래와 같이 나타낼 수 있다.

$$V_j^*(x_k) = \min_{u_j} [x_k^T S_j x_k + u_{j,k}^T R_j u_{j,k} + V_j^*(x_{k+1})]$$

이 때, (A_j, B_j)가 가안정성(stabilizable)이고, $(A_j, S_j^{1/2})$가 가검출(detectable)하면, 위 $V_j(x_k)$를 최소화시키는 안정한 최적의 $u_{j,k}^*$가 유일하게 존재한다. 이 때의 $V_j^*(x_k)$,$u_{j,k}^*$는 다음과 같이 나타낼 수 있다.

$$V_j^*(x_k) = x_k^T P_j x_k \qquad u_{j,k}^* = L_j x_k$$

여기서 P_j는 $n \times n$인 준양한정 행렬이며, 그 값은 다음 리카티 방정식(algebraic Riccati equation)의 해가 된다. 또한 L_j값은 아래와 같이 표현된다.

$$P_j = A_j^T P_j A_j + S_j - A_j^T P_j B_j [I + B_j^T P_j B_j]^{-1} [B_j^T P_j A_j]$$
$$L_j = (I + B_j^T P_j B_j)^{-1} (-B_j^T P_j A_j)$$

이제, 제안하는 알고리즘을 위한 j번째 하부시스템에 대한 Q-함수를 아래와 같이 정의하자.

$$Q_{j,k}(x_k, u_{j,k}) := x_k^T S_j x_k + u_{j,k}^T R_j u_{j,k} + x_{k+1}^T P_j x_{k+1} \quad (2)$$

식 (1)을 식 (2)에 대입하면 Q-함수를 아래와 같이 나타낼 수 있다.

$$Q_{j,k}(x_k, u_{j,k}) = \begin{bmatrix} x_k \\ u_{j,k} \end{bmatrix}^T \begin{bmatrix} S_j + A_j^T P_j A_j & B_j^T P_j A_j \\ A_j^T P_j B_j & R_j + B_j^T P_j B_j \end{bmatrix} \begin{bmatrix} x_k \\ u_{j,k} \end{bmatrix} := z_{j,k}^T H_j z_{j,k}$$

이 때, $H_j = \begin{bmatrix} H_{j,xx} & H_{j,xu} \\ H_{j,ux} & H_{j,uu} \end{bmatrix}$ 로 나타낼 수 있으며, H_j를 이용하여 앞에 결과를 아래와 같이 다시 나타낼 수 있다.

$$L_j = -(H_{j,uu})^{-1}(H_{j,ux}),\ P_j = [I\ L_j^T] H_j [I\ L_j^T]^T$$

이를 바탕으로 j번째 하부시스템에 대한 Q-학습법을 다음과 같이 유도할 수 있다 [4].

$$Q_{j,i+1}(x_k, u_{j,i}(x_k)) = x_k^T S_j x_k + u_{j,i}^T(x_k) R_j u_{j,i}(x_k) + Q_{j,i}(x_{k+1}, u_{j,i}(x_{k+1}))$$
$$L_{j,i} = -(H_{j,uu}^i)^{-1}(H_{j,ux}^i),\ u_{j,i}(x_k) = L_{j,i} x_k$$

여기서, $Q_{j,i}$는 $Q_{j,i} = [x_k^T\ u_{j,k}^T] H_{j,i} [x_k^T\ u_{j,k}^T]^T$이고 $H_{j,i}, u_{j,i}(x_k)$는 각각 j번째 하부시스템의 i번째 학습법을 통해 계산된 H행렬과 제어 입력이다. Q-학습법의 반복에 의해 $Q_{j,i}$와 $u_{j,i}$를 계산할 수 있으며, $i \rightarrow \infty$함에 따라, $Q_{j,i+1}(x_k, u_{j,i}(x_k)) \rightarrow Q_j^*(x_k, u_{j,k})$이고, 결국, $u_{j,i} \rightarrow u_j^*$가 되어, j번째 하부시스템의 최적 제어입력 u_j^*를 구할 수 있다 [4].

<표 1> Q-학습법 알고리즘

1: $i=0,\ P_{j,0}=0,\ H_{j,0}=0,\ {}^{\forall}j\in L=\{1,....,N\}$
2: $Z_{j,0}=0,\ Y_{j,0}=0,\ {}^{\forall}j\in L=\{1,....,N\}$
3: If subsystem l is active,
  if $j=l$
    $Z_j \leftarrow [Z_j\ \ \bar{z}(x_k)]$
    $Y_j \leftarrow [Y_j\ \ d(\bar{z}(x_k), h_{j,i})]$
  else
    $Z_j \leftarrow [Z_j],\ Y_j \leftarrow [Y_j]$
4: 최소자승법 계산이 가능하도록 Z_j, Y_j 행렬이 만들어졌는가?
5: $h_{j,i+1} \leftarrow (Z_j Z_j^T)^{-1} Z_j Y_j$
  $H_{j,i+1} \leftarrow f(h_{j,i+1})$, $h_{j,i+1}$로부터 $H_{j,i+1}$을 생성 [4].
  $L_{j,i+1} \leftarrow -(H_{uu})^{-1} H_{ux}$
6: $i \rightarrow i+1$, 3으로 돌아가서 활성화된 하부시스템에 맞게 반복 수행
7: 스위칭 변화가 없고, $\| h_{j,i+1} - h_{j,i} \|_F < \epsilon$ 이면 알고리즘 완료

표1은 위에서 설명한 알고리즘을 순서별로 정리한 것이다. 제안된 알고리즘의 순서는 다음과 같다. 우선 초기화 과정을 통해 모든 변수들을 0으로 만든다. 이 후 하부시스템 중 스위칭 신호에 의해 동작 중인 시스템에서는 Z_j, Y_j 데이터를 저장하여 행렬 H_j의 계산한다. Z_j, Y_j 행렬은 최소자승법을 이용하기 위해 계산하는데, 최소자승법이 사용된 이유는 하나의 식을 이용하여 $H_{j,xx}, H_{j,xu}, H_{j,ux}$ 행렬의 모든 성분들을 계산하기 위해서이다 [4]. 최소자승법을 수행할 수 있도록 충분한 Z_j, Y_j 행렬을 수집하고, 이를 통해 H_j행렬을 추정해 나가면서 각 하부시스템에 해당하는 최적의 제어 신호 u_j^*를 찾아 나간다. 제안된 알고리즘은 활성화된 하부시스템에서만 위의 계산을 수행하고 나머지의 경우에는 $Z_{j,i+1}=Z_i, Y_{j,i+1}=Y_i, H_{j,i+1}=H_i$ 와 같이 이전의 값을 저장한다.

3. 모의실험 결과

본 절에서는 모의실험을 통해 앞에서 제안된 알고리즘을 검증한다. 두 개의 하부시스템($N=2$)을 가지는 스위칭 시스템에 제안된 알고리즘을 수행하였고, 각 하부시스템의 모델은 다음과 같다.

$$A_1 = \begin{bmatrix} 0.9065 & 0.0816 & -0.0005 \\ 0.0074 & 0.9012 & -0.0007 \\ 0 & 0 & 0.1327 \end{bmatrix}, B_1 = \begin{bmatrix} -0.0015 \\ -0.0096 \\ 0.8673 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1.0000 & 0.0952 & 0.0045 \\ 0 & 0.9048 & 0.0861 \\ 0 & 0.0002 & 0.8187 \end{bmatrix}, B_2 = \begin{bmatrix} 0.0003 \\ 0.0091 \\ 0.1813 \end{bmatrix}$$

각각의 경우에 대한 리카티 방정식의 해 P_1, P_2와 그 때의 제어 값 L_1, L_2는 다음과 같다.

(A_1, B_1):

$$P_1 = \begin{bmatrix} 5.8229 & 2.6462 & -0.0039 \\ 2.6462 & 7.6004 & -0.0030 \\ -0.0039 & -0.0030 & 1.0101 \end{bmatrix},\ L_1 = [0.0197\ \ 0.0425\ \ -0.0661]$$

(A_2, B_2) :

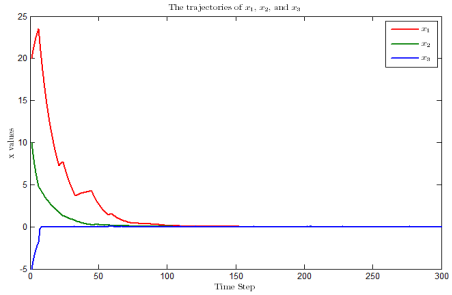
$$P_2 = \begin{bmatrix} 27.8752 & 17.3711 & 5.0273 \\ 17.3711 & 17.8997 & 5.0220 \\ 5.0273 & 5.0220 & 4.2173 \end{bmatrix},\ L_2 = [-0.9309\ \ -0.9315\ \ -0.6586]$$

본 모의실험에서는 스위칭 신호를 두 개의 하부시스템이 랜덤하게 활성화되도록 생성하였다. 그림 1는 제안된 알고리즘을 통해 계산된 x의 궤적을 나타낸 것이다. 그래프를 통해 알 수 있듯이 시간이 증가함에 따라 x값이 0에 점점 수렴함을 알 수 있다.

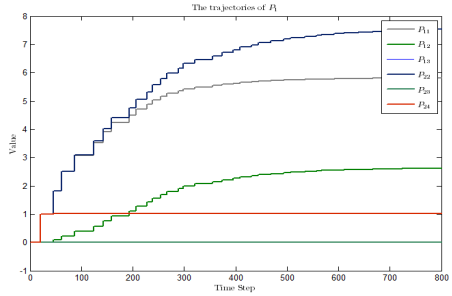
그림 2와 그림 3는 앞에서 제안된 알고리즘의 수행에 따라 나오게 된 P_1값과 L_2값의 궤적을 그래프로 나타낸 것이다. 시뮬레이션 결과를 통해 플랜트의 모델 정보가 없는 상황에서도 Q-학습법 알고리즘에 의해 각 값들이 앞에서 구한 최적의 P_1, L_2값으로 수렴함을 알 수 있다. 표2는 모의실험을 통한 최종 P_2, L_1값으로, 이 역시 최적의 값으로 수렴해 감을 알 수 있다.

<표 2> Q-학습법 알고리즘

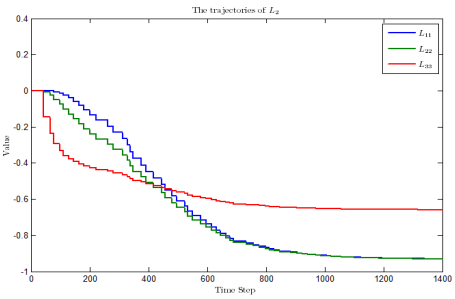
$P_2 = \begin{bmatrix} 27.8750 & 17.3709 & 5.0272 \\ 17.3709 & 17.8995 & 5.0219 \\ 5.0273 & 5.0219 & 4.2173 \end{bmatrix}$	$L_1 = [0.0197\ \ 0.0425\ \ -0.0661]$



<그림 1> 제어기를 통한 x의 궤적



<그림 2> $P_{1,i}$값의 궤적 변화



<그림 3> $L_{2,i}$값의 궤적 변화

4. 결 론

본 논문에서는 스위칭 선형 시스템의 적응 LQ 준최적 제어를 위한 Q-학습법을 제안하였다. 제안된 제어 알고리즘은 안정성이 증명된 Q-학습법에 기반하며 스위칭 시스템의 모델의 변수를 정확하게 모르는 상황에서도 준최적 제어가 가능하도록 설계되었다. 이 알고리즘을 기반으로 기존에 스위칭 시스템에서 고려하지 않았던 각 시스템의 불확실성 및 최적 적응 제어 문제를 해결하였고, 이를 모의실험을 통해 제안된 알고리즘의 성능과 결과를 검증하였다.

감사의 글 : 이 논문은 2011년도 두뇌한국21사업과 2010년도 지식경제부의 재원으로 한국에너지 기술평가원(KETEP)의 지원을 받아 수행한 연구 과제입니다. (No. 20104010100590)

[참 고 문 헌]

[1] J. Si, A. G. Barto, W. B. Powell and D. Wunsch, Handbook of Learning and Approximate Dynamic Programming, Wiley-IEEE Press, 2004.
[2] A. Tamimi, F.L. Lewis, M.A. Khalaf, "Model-free Q-learning Designs for Linear Discrete-time Zero-sum games with Application to H-infinity Control", Automatica, vol. 34, no. 3, pp 473-481, 2007.
[3] H. Lin and P. J. Antsaklis, "Stability and Stabilizability of Swiched Linear Systems: A Survey of Recent Results", IEEE Trans. on Automatic Control, vo.l 54, no. 2, 2009.
[4] F. L. Lewis and D. Vrabie "Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control", IEEE Circuits and Systems Magazine, Third Quarter, 2009.