퍼지 LQRQL 제어

Fuzzy LQRQL Control

김영일, 김종호, 박주영

Young-Il Kim, Jongho Kim, Jooyoung Park

고려대학교 제어계측공학과

Dept. of Control and Instrumentation Engineering

Korea University

E-mail : oyeasw@korea.ac.kr

요 약

Q-learning은 강화학습의 한 방법으로서, 여러 분야에 널리 응용되고 있는 기법이다. 최근에는 Linear Quadratic Regulation (이하 LQR) 문제에 성공적으로 적용된 바 있다. 특히 시스템 모델의 파라미터에 대한 구체적인 정보가 없는 상태에서 적절한 입력과 출력만을 가지고, 학습을 통해 문제를 해결할 수 있어서 상황에 따라서 매우 실용적인 대안이 될 수 있다. 이에 따라 본 논문에서는 이러한 일반적인 LQR Q-learning(이하 LQRQL) 학습방법에 퍼지 모델을 이용하여 제어기를 설계하는 방법을 고려하고, 일반적인 LQRQL 기법과 본 논문에서 제시한 방법의 결과를 비교하여 응용 가능성을 살펴보았다.

1.서론

강화 학습은 모델-기반 방법과 모델을 필요로 하지 않는 방법으로 구분되는데, 큐 학습은 후자에 속한 방법이다. 큐 학습에서는 미래에 받게 되는 강화신호(reinforcement signal)의 총합을 상태와 입력에 관한 함수로 근사화 되는데, 이 함수가 바로 큐 함수(Q function)이다. 이런 큐 함수를 이용하면 시스템 파라미터를 구체적으로 알지 못하는 상태에서도 관찰된 비용 값을 이용한 학습을 통해서 최적 제어 입력을 구할 수 있다. 본 논문에서는 큐 함수 에 퍼지 멤버쉽 함수 (membership function)를 이용한 개선된 제어방법을 새로이 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 일반적인 LQRQL 학습방법에 대하여 설명하고, 3장에서는 퍼지 모델을 이용한 LQRQL 학습 방법을 설명한다. 4장에서는 예제를 통한 모의실험을 통해 3장에서 제시한 LQRQL 방법에 퍼지 모델을 이용한 학습 방법의 결과를 살펴본다. 5장에서는 결론을 제시하고 향후 연구 방향과 과제에 대해서 논의한다.

2. LQRQL 방법

2.1 LQR

LQR의 기본 구조는 선형적 특성을 갖는 시스템과 2차의 형태를 갖는 손실로 이루어진다. 선형 시불변 이산시간(linear time invariant discrete time) 시스템은 다음과 같이 정의된다.

$$x_{k+1} = Ax_k + Bu_k \tag{1}$$

손실 값인 r_k는 k스텝에서 상태와 제어 입력의 2차 함수의 형태로 이루어져 있으며 이는 다음 식으로 나타낼 수 있다.

$$r_k(x_k, u_k) = x_k^T P x_k + u_x^T R u_k \tag{2}$$

여기에서 $P \in R^{n \times n}$ 와 $R \in R^{p \times p}$는 설계자가 선택하게 되는 대칭 행렬인데, 일반적으로 양의 정부호(positive definite)임이 가정된다. 결국 우리의 목표는 다음 식에 손실의 총합 값으로 나타낸 J를 최소화시키는 제어규칙을 찾는 것이다.

$$J = \sum_{k=0}^{\infty} r_k \tag{3}$$

2.2 Q 함수

모델이 필요 없는 강화학습 방법은 상태변화의 확률값인 $P_{ss'}^{a}$와 강화의 기댓값인 $R_{ss'}^{a}$의 값을 사용하지 않는다.[2][3] Q 학습에서는 피드백이 상태와 행동으로 표현된 형태로 Q 함수로부터 얻어진다. 즉 제어 입력 값을 최소로 할 수 있는 Q 함수 값을 찾는 것이 되며, 이것을 식으로 나타내면 다음과 같다.

$$V^*(s) = \min_u Q^*(x, u) = Q^*(x, u^*) \tag{4}$$

식 (4)을 정리해 보면 다음과 같은 식으로 나타난다[1]

$$\begin{aligned} & Q^*(x_k, u^*_k) \\ &= \begin{pmatrix} x_k^T & u_k^{*T} \end{pmatrix} \begin{pmatrix} P + A^T K^* A & A^T K^* B \\ B^T K^* A & R + B^T K^* B \end{pmatrix} \begin{pmatrix} x_k \\ u^*_k \end{pmatrix} \\ &= \begin{pmatrix} x_k^T & u_k^{*T} \end{pmatrix} \begin{pmatrix} H^*_{xx} & H^*_{xu} \\ H^*_{ux} & H^*_{uu} \end{pmatrix} \begin{pmatrix} x_k \\ u^*_k \end{pmatrix} \\ &= \phi_k^{*T} H^* \phi_k^* \end{aligned} \tag{5}$$

우리는 식 (5)을 이용하여 시스템의 모델을 알 수 없는 상태에서 최적의 제어 입력 값을 계산해 낼 수 있다. 그리고 우리가 원하는 피드백 값인 최적의 제어 입력 값은 아래와 같이 표현가능 하다.

$$u^*_k = \arg\min_u Q^*(x_k, u) \tag{6}$$

식 (5)에서 Q-함수가 2차식의 형태를 갖고있고, H^*가 0값 보다 큰 대칭행렬(symmetric positive definite matrix)의 형태로 표현되는 것을 알 수 있다. 그래서 식 (5)를 제어 입력 값에 대해서 미분하여 그 값을 0 으로 하는 값으로 (6)의 결과를 구할 수 있다. 이를 식으로 나타내면 다음과 같다.

$$\nabla_{u_i} Q^*(x_k, u^*_k) = 2H^*_{ux} x_k + 2H^*_{uu} u^*_k = 0 \tag{7}$$

식 (7)의 결과로 다음과 같은 값을 얻게 된다.

$$u^*_k = -(H^*_{uu})^{-1} H^*_{ux} x_k = L^* x_k \;,\; L^* = -(H^*_{uu})^{-1} H^*_{ux} = B^T K^* A \quad \text{and} \quad H^*_{uu} = R + B^T K^* B \tag{8}$$

식 (8)이 우리가 얻고자 했던 최적의 제어 입력 값이다.

2.3 LQRQL 제어

LQR 문제의 목적은 총 비용 $\sum_{k=0}^{\infty} r_k$ 를 최소화 시키는 제어규칙 $g^*: R^n \rightarrow R^p$, 즉 최적 제어 $u = g^*(x)$ 를 찾는 것이다. 그런데, 정해진 상태 궤환 제어 $g(x) = Lx$ 에 대하여 큐 함수 $Q(x, y) = Q^L(x, y)$를 전개하면, 양의 정부호인 이차형식이 되므로[4], 큐 함수는 다음과 같이 양의 정부호 행렬 H^L를 파라미터로 갖는 이차 형식으로 표현될 수 있다.

$$\begin{aligned} Q^L(x_k, u_k) &= \sum_{i=k}^{\infty} r_i = r_k + \sum_{i=k+1}^{\infty} r_i \\ &= r_k + Q^L(x_{k+1}, Lx_{k+1}) \end{aligned} \tag{9}$$

식 (9)는 또한 다음과 같이 나타낼 수도 있다.

$$r_k + Q^L(x_{k+1}, Lx_{k+1}) - Q^L(x_k, u_k) = 0$$

$$\begin{aligned} r_k &= Q^L(x_k, u_k) - Q^L(x_{k+1}, Lx_{k+1}) \\ &= \phi_k^T H^L \phi_k - \phi_{k+1}^T H^L \phi_{k+1} \end{aligned} \tag{10}$$

$$\phi_k^T = [x_k^T \; u_k^T], \quad \phi_{k+1}^T = [x_{k+1}^T \; L^T x_k^T]$$

식 (10)에서 Q-함수를 상태를 나타내는 스칼라의 x와 제어입력을 나타내는 스칼라의 u 를 이용하면 Q 함수를 다음과 같이 나타낼 수 있다[4]

$$\begin{aligned} Q(x_k, u_k) &= \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \begin{bmatrix} h_{xx} & h_{xu} \\ h_{ux} & h_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &= \begin{bmatrix} h_{xx} & h_{xu} + h_{ux} & h_{uu} \end{bmatrix} \begin{bmatrix} x_k^2 \\ x_k u_k \\ u_k^2 \end{bmatrix} \\ &= \theta^T \xi = Q(\xi) \end{aligned} \tag{11}$$

3. 퍼지 LQRQL 제어

퍼지 LQRQL 학습방법은 퍼지 모델을 이용하여, 퍼지 멤버쉽 함수의 형태로 상태 공간을 나누어 제어입력을 구하게 된다.

퍼지 멤버쉽 함수 $\Phi(a)$를 사용하여 식 (11)에 나타낸 Q-함수의 값을 다음과 같이 정의한다

$$Q(x_k, u_k) = \sum_{i=1}^{3} \Phi_i(a) \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \begin{bmatrix} h_{xx} & h_{xu} \\ h_{ux} & h_{uu} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

$$= \sum_{i=1}^{3} \Phi_i(a)\ Q(\xi) \qquad (12)$$

그리고, 식 (11)에서 나타낸 Q-함수의 값을 이용하면, 식 (10)는 다음과 같이 나타낼 수 있다.

$$r_k = Q^L(x_k, u_k) - Q^L(x_{k+1}, Lx_{k+1})$$

$$= \sum_{i=1}^{3} \Phi_{i(a)}\ \phi_k^T H^L \phi_{k-} \sum_{i=1}^{3} \Phi_i(a)\ \phi_{k+1}^T H^L \phi_{k+1}$$

$$= \sum_{i=1}^{3} \Phi_i(a)[Q_k(\xi) - Q_{k+1}(\xi)] \qquad (13)$$

즉, 퍼지 모델의 멤버쉽 함수(membership function)를 이용한 형태의 LQRQL 제어기 모델을 얻게 된다. 식 (13)에서 Q-함수를 새롭게 정의하기 위하여 사용된 퍼지 함수 $\Phi(a)$는 세 가지의 형태로, 그림 1 형태의 퍼지 모델과 사다리꼴 형태의 퍼지 모델(그림 2), 그리고 일반화시킨 형태의 가우시안 퍼지 모델(그림 3)을 사용하였다.

4. 모의실험결과

4.1 모의실험에 사용된 비선형 시스템 -이동 로봇(The mobile robot)의 모델

실험에 사용된 로봇의 위치는 직선에 대한 거리 δ로 나타내고, 직선에 대한 각도는 α로 나타낸다. 거리는 'meter'를, 각도에는 'radian'을 기본단위로 선택하였다. 기하학적으로 로봇의 무게중심은 두 바퀴의 중앙에 위치한다. 실험에서 제어 입력은 직선으로 움직이는 속도에 대한 v_t 와 회전 속도에 대한 w로 나타낸다.

$$\alpha_{k+1} = \alpha_k + wT$$

$$\delta_{k+1} = \delta_k + \frac{v_t}{w}(\cos(\alpha_k) - \cos(\alpha_k + Tw)) \qquad (14)$$

$$where,\quad w = Lx_1\ ,\ x_1 = \alpha$$

$x^T = [\alpha\ \ \delta]$라 하고, 2차식 형태의 손실 값을 나타내는 함수는 다음과 같이 정의하였다.

$$r_k = x_k^T \begin{pmatrix} S_\alpha & 0 \\ 0 & S_\delta \end{pmatrix} x_k + u_k R u_k \qquad (15)$$

$$where\ \ S_\alpha = 0.1\ ,\ S_\delta = 1\ ,\ R = 1 \text{ and } u = w$$

4. 2. 모의실험 결과

제안한 방법으로 설계된 비선형 제어기를 앞에서 설명한 로봇의 위치 제어 문제에 적용하고 결과를 비교하였다. 예제에 사용된 로봇의 학습 데이터는 초기 위치 네 곳의 d_0에서 시작하여 이동한 위치 데이터로써, 로봇은 각기 영점에서 d_0만큼 떨어진 거리에서 출발하여 좌표 상으로 영점에서부터 수평방향으로 무한한 길이를 갖는 직선을 따라가는 것을 목표로 하여 두 가지 종류의 시뮬레이션 실험 데이터를 발생시키고, 이 데이터를 가지고 실험하였다.

첫 번째 실험 초기위치→

$$d_0 = [1.5, 0.75, -0.75, -1.5]$$

두 번째 실험 초기위치→

$$d_0 = [1.8,\ 1.0,\ -1.0, -1.8]$$

모의실험에서는 로봇의 각도 α에 대하여 퍼지 멤버쉽 함수를 적용하여 상태공간을 구분하고 제어기를 설계하였다. 그리고 각각의 데이터에 대하여 2회 반복 학습하여 그 결과를 확인하였다. 그림은 제어기 설계에 사용한 퍼지 멤버쉽 함수의 형태를 나타낸 것이다.

표 (1)과 (2)는 LQRQL 제어기와 퍼지 LQRQL 제어기를 적용한 결과 나타난 총 손실의 값을 비교한 표이다.

2회 반복 학습의 결과		로봇의 출발 위치			
		1.5	0.75	-0.75	-1.5
COST	LQRQL	56.96	13.54	13.54	56.96
	퍼지 모델	35.39	10.97	10.56	35.37
	사다리꼴 형태	34.96	11.44	10.36	35.78
	일반화시킨 RBF	35.12	11.21	10.66	34.64

표 1 첫 번째 데이터에 대한 LQRQL 제어기와 퍼지 LQRQL 제어기의 총 손실 값 비교

2회 반복 학습의 결과		로봇의 출발 위치			
		1.8	1.0	-1.0	-1.8
COST	LQRQL	84.07	24.19	24.19	84.07
	퍼지 모델	49.73	17.49	16.86	49.11
	사다리꼴 형태	49.03	17.15	16.90	48.94
	일반화시킨 RBF	49.36	17.24	17.70	49.03

표 2 두 번째 데이터에 대한 LQRQL 제어기와 퍼지 LQRQL 제어기의 총 손실 값 비교

이상의 관찰로부터, 본 논문의 퍼지 LQRQL 제어기를 적용한 학습방법은 일반적인 LQRQL 제어기를 사용한 방법과 비교하여 2회 반복하여 학습한 결과 손실 값을 상당히 줄일 수 있음을 알 수 있다.

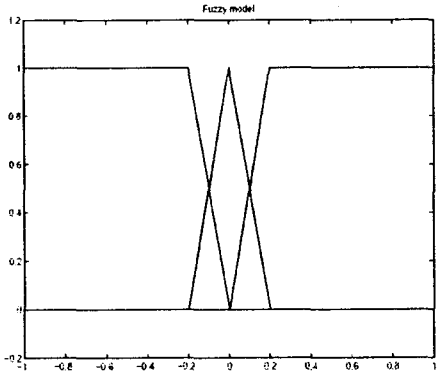

그림 1 예제에 적용된 일반적인 형태의 퍼지 모델

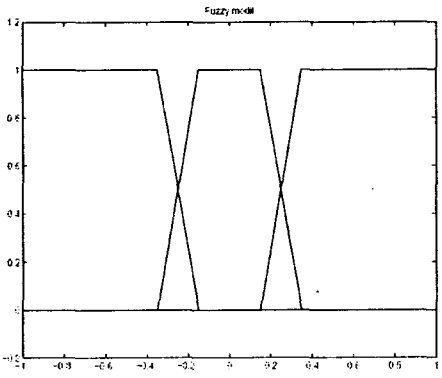

그림 2 예제에 적용된 사다리꼴 퍼지 모델

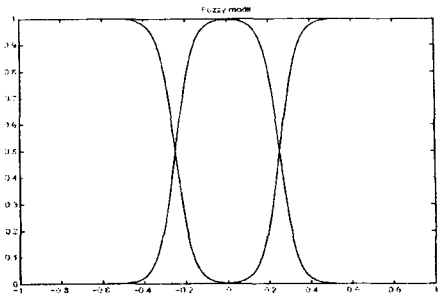

그림 3 예제에 적용된 일반화시킨 가우시안 모델

5. 결 론

본 논문에서는 일반적인 LQRQL 학습 방법을 전반적으로 소개한 후에, 기존의 방법론을 참고하여 퍼지 모델을 이용한 퍼지 LQRQL 제어기 설계 기법을 제시하고 필요한 중간 과정을 유도하여 보았다. 관련 연구에서 고려된 비선형 시스템인 로봇의 위치제어 문제를 대상으로 비교·검토해 본 결과, 주어진 로봇의 위치제어 문제에서 본 논문에서 제시한 방법론이 기존의 LQRQL 학습 방법보다 더 좋은 결과를 제공함을 관찰할 수 있었다. 이와 관련하여 이후에 추가적으로 연구되어져야 할 과제로는 학습이 2회 이상 반복될 경우에도 성능이 꾸준히 향상될 수 있도록 보장하는 학습 메카니즘을 개발해야 해야하는 과제가 있다. 그리고 다양한 예제에 대한 광범위한 실험을 통하여 제시한 방법론의 장단점을 파악하고, 여타 방법론과의 체계적인 성능 비교가 필요하다.

참고문헌

[1] S. Hagen and B. Kröse. "Linear quadratic regulation using reinforcement learning," *In Proc. of the 8th Belgian-Dutch Conf. on Machine Learning*, 1998.

[2] C. Watkins. *Learning from Delayed Rewards*. PhD Thesis, University of Cambridge, 1989.

[3] C. Watkins and P. Dayan. "Technical note: Q-learning," *Machine Learning*, 1992.

[4] S. Hagen. *Continuous State Space Q-learning for Control of Nonlinear Systems*, PhD Thesis, Computer Science Institute, University of Amsterdam, 2001.