마코프 의사 결정 과정에 대한 제어

■ 한수희
(서울대학교 공과대학 기계 · 항공공학부)

1. 서 론

확률 동적 자원 분배 문제는 시스템을 효과적으로 구현하고 운영하기 위하여 중요하게 연구되고 있다. 어떤 주어진 일들을 효과적으로 처리하기 위해서는 시스템의 자원을 효율적으로 분배 하고 관리할 수 있도록 시스템 안의 제어기를 잘 설계해야 한다. 잘 설계된 제어기는 시스템의 성능이 최적화 되도록 나름대로의 판단력을 갖추어 의사를 결정한다. 이러한 의사 결정 제어기를 구현하기 위한 수학적인 모델로 마코프 의사 결정 과정 모델이 있다. 이 모델에서는 정해진 의사 결정에 시스템이 확률적으로 동작한다. 시스템의 상태 정보를 바탕으로 의사 결정을 하고 그 것에 따른 보상을 받는다. 무한 구간에서 시스템의 확률적 변화에 대한 보상의 기대치를 최대화하는 의사 결정 제어기를 설계하는 것이 중요하다. 이러한 의사 결정 제어기는 많은 분야에 걸쳐 응용이 되고 연구가 진행되고 있다.

본고에서는 마코프 의사 결정 모델을 알아보고 최적화를 하는 방법을 학습한다. 또한 이미 연속형 변수의 상태 방정식에 익숙한 제어 전공자들이 쉽게 이해할 수 있도록 선형 이차 최적 제어(Linear Quadratic Control, LQC)를 예로 들어 설명한다.

2. 마코프 의사 결정 모델

마코프 의사 결정 모델은 그림 1과 같이 구성된다 [1].

플랜트는 유한한 상태를 갖고 이산 시간으로 동작하는 마코프 사슬이고, 제어기는 플랜트의 상태에 따라서 보상을 최적화하도록 적절한 제어(control, action)를 취한다. 이 제어에 따라서 마코프 사슬의 전이 확률은 바뀌고 보상에 영향을 준다. 제어값으로 a가 주어졌을 때 상태 i 에서 j 로의 전이 확률은 마코프 성질을 고려하여 다음과 같이 표현된다.

$$P_{ij}(a) = P(X_{n+1} = j \mid X_n = i,\ A_n = a) \tag{1}$$

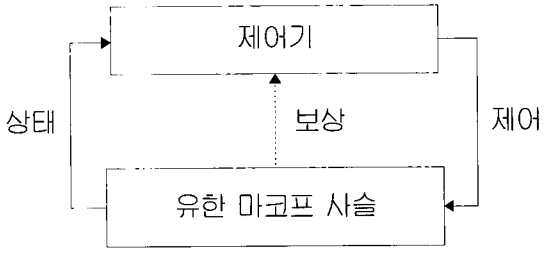

그림 1. 마코프 의사 결정 모델

여기서 X_{n+1}, X_n, A_n은 시간 $n+1$에서의 상태, 시간 n에서의 상태와 제어 값을 각각 나타내는 확률변수이다. 본고에서는 N개의 상태를 가정하고 각각 $1,2,\cdots,N$의 지수로 표시한다. 우리의 관심 사항은 제어 값 $a_k(k=1,\cdots,K-1)$들을 다음과 같은 가격함수가 최소화 되도록 결정하는 것이다.

$$J_0(X_0)=E[g_K(X_K)+\sum_{k=0}^{K-1}g_k(X_k,\ a_k)] \tag{2}$$

여기서 $g_k(\cdot,\cdot)$는 항상 양의 값을 가지며, 위로 유계인 함수이다. 연속형 변수를 다루는 상태 방정식에서 최적 제어가 주로 상태 궤환 형태로 나오듯이 최적의 제어 값 a_k도 상태의 함수로 표현된다. 즉 $\mu(X_k)$와 같이 쓸 수 있다. 상태를 제어 값으로 바꾸어 주는 함수 $\mu(\cdot)$를 가격 함수 (2)가 최소화 되도록 구하는 것이 우리의 목표이다.

3. 동적 프로그래밍 알고리즘과 Bellman의 최적 조건식

3.1 동적 프로그래밍 알고리즘

최적의 경로는 그 중간의 지점부터 동일한 종점까지의 최적성도 보장한다. 이 원리를 최적의 원리(principle of optimality)라고 부르며 동적 프로그래밍의 근본 개념이기도 하다. 동적 프로그래밍은 아래와 같이 표현할 수 있다.

$$\begin{aligned}J_n^*(X_n)=&\min_{\mu_n}E[g_n(X_n,\mu_n(X_n))]\\&+J_{n+1}^*(X_{n+1})]\end{aligned} \tag{3}$$

여기서 $J_{n+1}^*(X_{n+1})$은 다음과 같이 주어지며,

$$\begin{aligned}J_{n+1}^*(X_{n+1})=&\min_{\mu_k}E[g_K(X_K)\\&+\sum_{k=n+1}^{K-1}g_k(X_k,\mu_k(X_k))]\end{aligned} \tag{4}$$

경계 값은 $J_K^*(X_K)=g_K(X_K)$으로 설정한다. 모든 $\mu_n^*(n=0,\cdots,K-1)$이 (3)식의 우변 식을 최소화 시키면, 생성된 제어 값 $\mu_0^*,\mu_1^*,\cdots,\mu_{K-1}^*$은 최적의 상태 경로를 생성한다. 관계식 (3)에 의하면 마지막 단의 최적화 문제를 풀고, 그 것을 바탕으로 마지막 두 단계의 최적화 문제를 푸는 식으로 모든 단계의 최적화 문제를 푼다. 최적의 원리에 의해 이후 단계의 최적성은 이전 단계에서도 보존되므로 동적 프로그램은 최종적으로 최적의 해를 제공한다는 것을 알 수 있다.

상태 방정식의 LQC를 유도하는 경우에도 동적 프로그래밍은 유용하게 쓰일 수 있으며, 특히 추적 LQC의 경우 변분법과 같은 다른 방법보다 더 간단하게 해와 최적의 가격함수 값을 유도할 수 있다[2][3].

2. Bellman의 최적 조건식

바로 전에 설명한 동적 프로그래밍은 기본적으로 유한 구간의 최적화 문제를 푸는데 적합하다. 이 기법을 확장하여 이번에는 무한 구간의 최적화 문제를 푸는 방법을 소개한다. 또한 시간마다 제어 값이 변하는 것이 아닌 고정된 경우에 대해서 다룬다. 여기에서는 가격함수 (2)의 $g_k(\cdot,\cdot)$을 $\gamma^k g(\cdot,\cdot)$으로 놓고, 시간 순서를 역으로 바꾸어서 다음과 같은 동적 프로그램을 생각해 보자.

$$J_{n+1}^*(X_0)=\min_{\mu}E_{X_1}[g(X_0,\mu(X_0))+\gamma J_n^*(X_1)] \tag{5}$$

여기서 $J_n^*(X_0)$는 다음과 같이 주어진다.

$$J_n^*(X_0)=\min_{\mu}\sum_{j=0}^{n}\gamma^j g_j(X_j,\mu(X_j)) \tag{6}$$

초기 상태 $X_0=i$에 대해 무한 구간의 최적 가격 $J^*(i)$는 유한 구간의 최적 가격 $J_n^*(i)$의 극한으로 생각해 볼 수 있으므로 모든 상태 i에 대해 다음과 같다.

$$J^*(i)=\lim_{n\to\infty}J_n^*(i) \tag{7}$$

따라서 동적 프로그래밍 (5)는 다음과 같이 나타낼 수 있다.

$$J^*(i)=\min_{\mu}E_{X_1}[g(i,\mu(i))+\gamma J^*(X_1)] \tag{8}$$

위의 기대 값을 계산하면 아래와 같은 식을 얻는다.

$$J^*(i)=\min_{\mu}[c(i,\mu(i))+\gamma\sum_{j=1}^{N}p_{ij}(\mu)J^*(j)] \tag{9}$$

여기서 $c(i,\mu(i))$는 아래와 같이 주어진다.

$$c(i,\mu(i))=\sum_{j=1}^{N}p_{ij}(\mu)g(i,\mu(i)) \tag{10}$$

관계식 (9)를 Bellman의 최적화 식이라고 부른다. 이 식은 점화적인 표현이 아니고 N개의 방정식으로 이루어졌고, 각각의 해 $(J^*(1),J^*(2),\cdots,J^*(N))$는 각 상태에서의 최적 가격함수를 말

한다. 다음 두 장에서는 Bellman의 최적화 (9)를 풀어 최적의 해를 얻기 위한 방법을 소개한다.

4. 제어 반복법을 통한 최적화

제어 반복법을 소개하기 이전에 Q factor라는 개념을 먼저 설명한다. 기존의 제어 μ가 있고 해당 가격함수 $J^{\mu}(i)$가 모든 상태 i에 대하여 알려져 있다고 가정하자. 상태 i와 제어 값 a에 대한 Q factor는 따라오는 가격함수와 현재의 가격을 더한 값으로 정의한다.

$$Q^{\mu}(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^{\mu}(j) \tag{11}$$

Q factor가 제어 μ에 의해 다음과 같이 정해질 때 제어 μ를 greedy하다고 한다.

$$Q^{\mu}(i,\mu(i)) = \min_a Q^{\mu}(i,a) \tag{12}$$

Q factor와 greedy 제어 개념을 사용하여 제어 반복법을 사용한 가격함수 최적화에 대하여 설명한다. 이 방법은 다음의 두 가지 과정을 거친다.

1. **제어 평가 단계 :** 현재 제어에 대한 가격함수와 모든 상태와 제어 값에 대한 해당 Q factor를 계산한다.
2. **제어 개선 단계 :** 1 단계에서 계산된 Q factor에 대해 개선되도록 현재의 제어를 갱신한다.

현재의 제어 μ_n가 주어졌을 때 제어 평가 단계에서는 아래의 선형 방정식을 풀어

$$J^{\mu_n}(i) = c(i,\mu_n(i)) + \gamma \sum_{j=1}^{n} p_{ij}(\mu_n(i)) J^{\mu_n}(j) \tag{13}$$

가격함수 $J^{\mu_n}(1)$, $J^{\mu_n}(2)$, $\cdots$, $J^{\mu_n}(N)$을 구한다. 구한 가격함수를 사용하여 (11)식과 같이 Q factor를 계산할 수 있다. 제어 개선 단계에서는 새로운 제어를 아래의 최적화 문제로부터 구한다.

$$\mu_{n+1}(i) = \arg\ \min_a Q^{\mu_n}(i,a) \tag{14}$$

여기서 $i = 1,\cdots,N$이다. (13)과 (14)에 의해 반복해서 얻어진 제어에 해당하는 가격함수는 단조 감소함을 알 수 있는데, 이것은 아래와 같이 증명할 수 있다. 우선 고정된 n에 대하여 다음과 같은 점화식을 생각한다.

$$\begin{aligned} J^{(m+1)}(i) &= c(i,\mu_{n+1}(i)) \\ &\quad + \gamma \sum_{j=1}^{n} p_{ij}(\mu_{n+1}(i)) J^{(m)}(j) \end{aligned} \tag{15}$$

초기값은 $J^{(0)}(i) = J^{\mu_n}(i)$로 주어지고, $m = 0,1,\cdots$이다. 점화 (15)로부터 다음과 같은 부등식을 얻는다.

$$\begin{aligned} J^{(0)}(i) &= c(i,\mu_n(i)) + \gamma \sum_{j=1}^{n} p_{ij}(\mu_n(i)) J^{(0)}(j) \\ &\geq c(i,\mu_{n+1}(i)) + \gamma \sum_{j=1}^{n} p_{ij}(\mu_{n+1}(i)) J^{(0)}(j) \\ &= J^{(1)}(i) \end{aligned} \tag{16}$$

동적 프로그래밍의 단조성 성질 때문에 (12)로부터 생성된 $J^{(m)}(i)$은 다음을 만족한다.

$$J^{(0)}(i) \geq J^{(1)}(i) \geq J^{(2)}(i) \cdots \tag{17}$$

$n \to 0$에 따라 $J^{(n)}(i) \to J^{\mu_{n+1}}(i)$이기 때문에 다음과 같은 결과를 얻는다.

$$J^{\mu_n} = J^{(0)} \geq J^{\mu_{n+1}} \tag{18}$$

즉 J^{μ_n}이 항상 양이고, n에 대해 단조 감소함으로 수렴함을 알 수 있다. 따라서 위의 두 가지 단계(제어 평가 및 개선 단계)를 아래와 같은 조건이 나타날 때까지 수행한다.

$$J^{\mu_{n+1}} = J^{\mu_n} \tag{19}$$

마코프 의사 결정 모델이 유한한 개수의 상태를 갖고 있기 때문에 유한한 횟수의 반복 후에 (19)가 성립할 것이다.

5. 가격함수의 반복을 통한 최적화

제어 반복법인 경우는 가격함수를 매번 다시 계산해야하는 부담이 있다. 아번에 소개되는 방법은 가격함수를 반복 근사화함으로써 한 번에 가격함수를 대수방정식을 통해 풀어야 하는 부담을 없앴다.

$J_n(i)$는 상태 i에서 n번 반복을 통해서 구한 가격함수라고

하자. 이 알고리즘은 임의로 설정한 초기 값 $J_0(i)$에서 시작한다. 일단 $J_0(i)$가 결정됐으면 아래 식으로부터 $J_1(i)$, $J_2(i),\cdots$를 구한다.

$$J_{n+1}(i) = \min_a c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j) \qquad (20)$$

위와 같이 계속 반복을 하면 가격함수는 최적의 해 $J^*(i)$에 수렴해 간다. 충분히 수렴된 $J_n(i)$ 값을 바탕으로 아래와 같이 최적 제어를 구한다.

$$\mu_{n+1}(i) = \arg\ \min_a Q^*(i,a) \qquad (21)$$

여기서 $Q^*(i,a)$ 은 다음과 같이 주어진다.

$$Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j)$$

위에서 설명한 가격함수의 반복을 통한 최적화를 정리하면 아래와 같다.

1. 상태 $i = 1,2,\cdots,N$ 에 대하여 초기 가격함수 $J_0(i)$ 를 결정한다.
2. $n = 0,1,\cdots$ 에 대해 (20)의 과정을 다음의 식이 만족될 때 까지 수행한다.

$$|J_{n+1} - J_n| < \epsilon \qquad (22)$$

여기서 ϵ 은 양의 수로 주어지는 설계 파라미터이다. ϵ 이 충분히 작으면 $J_n(i)$이 최적의 $J^*(i)$에 충분히 가깝다고 생각하자. 다음과 같이 Q factor를 계산할 수 있다.

$$Q^*(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J_n(j) \qquad (23)$$

greedy 알고리즘에 의해 최적의 제어를 다음과 같이 구한다.

$$\mu^*(i) = \arg\ \min_a Q^*(i,a) \qquad (24)$$

제어 반복법을 통해서 최적 해를 찾는 경우는 유한한 반복만 필요하지만, 가격함수 반복법을 통한 최적화는 무한의 반복이 요구될 수 있다. 이유는 제어 반복법인 경우 반복 시에 동일한 제어기가 생성되지 않기 때문이다.

6. 제어와 가격함수의 반복을 사용한 LQ 최적 제어의 계산

마코프 의사 결정 모델을 바탕으로 설명한 최적화의 방법은 연속형 변수의 상태 공간에서도 적용될 수 있다. 기존의 상태 방정식에 익숙한 제어 전공의 독자들이 쉽게 이해할 수 있도록 무한 구간의 LQ 최적 제어를 위에서 설명한 두 가지 방법에 의해 수치 해석적으로 계산해 본다. 또한 잡음 등을 고려한 stochastic 시스템이 아니라 deterministic 시스템을 다루어 계산 과정을 좀 더 쉽게 알 수 있도록 한다.

우선 다음과 같은 상태 방정식을 생각하자.

$$x_{k+1} = Ax_k + Bu_k \qquad (25)$$

여기서 A와 B는 아래와 같이 주어진다.

$$A = \begin{bmatrix} 0.17 & 0.36 & 0.98 \\ 0.25 & 0.85 & 0.85 \\ 0.10 & 0.92 & 0.76 \end{bmatrix},\quad B = \begin{bmatrix} 0.86 \\ 0.28 \\ 0.09 \end{bmatrix}$$

rank $[B\ \ AB\ \ A^2B] = 3$이므로 시스템 (25)는 가제어성임을 알 수 있다. 초기 상태 x_0가 주어질 때 시스템 (25)의 입력 u_i를 아래의 가격함수가 최소가 되도록 구한다.

$$J^u(x_0) = \sum_{i=0}^{\infty} x_i^T Q x_i + u_i^T R u_i \qquad (26)$$

여기서 Q 행렬은 양의 반한정으로 R 행렬은 양의 한정으로 주어진다. 가격함수 (26)을 보면 (9)의 $c(\cdot,\cdot)$ 가 다음과 같이 주어졌음을 알 수 있다.

$$c(i,u_i) = x_i^T Q x_i + u_i^T R u_i \qquad (27)$$

최적 제어 u_i는 리카티 방정식의 해를 이용해서 상태 궤환 형태로 주어짐이 잘 알려져 있는데, 여기서는 정확한 최적 해를 유도하지 않고, 위에서 설명한 반복 방법을 사용하여 수치 해석적으로 해를 계산하는 방법에 대해 소개할 것이다.

6.1 제어 반복법을 이용한 최적 제어 구하기

초기 가격함수 $J^u(x) = x^T P^{(0)} x$와 초기 제어 $u = K^{(0)} x$가 주어진다면, (13)에 해당하는 관계식을 다음과 같이 만들 수 있다.

$$x_i^T P^{(n)} x_i = x_i^T Q x_i + u_i^T R u_i + (Ax_i + Bu_i)^T P^{(n)} (Ax_i + Bu_i) \quad (28)$$

여기서 n 번 반복한 가격함수와 제어를 $x_i^T P^{(n)} x_i$ 와 $K^{(n)} x_i$ 로 표시했다. (28)이 x_i 에 상관없이 항상 성립하도록 아래의 $P^{(n)}$ 에 관한 대수 행렬 방정식을 구한다.

$$P^{(n)} = Q + R + (A + BK^{(n)})^T P^{(n)} (A + BK^{(n)}) \quad (29)$$

수식 (29)는 고정된 $K^{(n)}$ 에 대해 이산형 리아푸노프 방정식이므로 CEMTool[4] 또는 MATLAB[5]의 함수 "dlyap"를 사용하면 쉽게 구할 수 있다. 구해진 $P^{(n)}$ 에 대하여 제어 개선 단계에서는 다음과 같이 새로운 u_i 를 계산한다.

$$\min_{u_i} x_i^T Q x_i + u_i^T R u_i + x_{i+1}^T P^{(n)} x_{i+1} \quad (30)$$

x_{i+1} 에 $Ax_i + Bu_i$ 를 대입하면 u_i 에 대한 이차 형식의 식을 얻는다. u_i 에 대한 미분치를 0 으로 놓고 u_i 를 구하면 u_i 는 아래와 같이 표현된다.

$$u_i = K^{(n+1)} x_i = -(R + B^T P^{(n)} B)^{-1} B^T P^{(n)} A x_i \quad (31)$$

위와 같은 반복을 수행하여 아래와 같이 $K^{(n)}$ 와 $P^{(n)}$ 를 구하면

$$K^{(0)} \rightarrow P^{(0)} \rightarrow K^{(1)} \rightarrow P^{(1)} \cdots \rightarrow K^{(n)} \rightarrow P^{(n)} \cdots$$

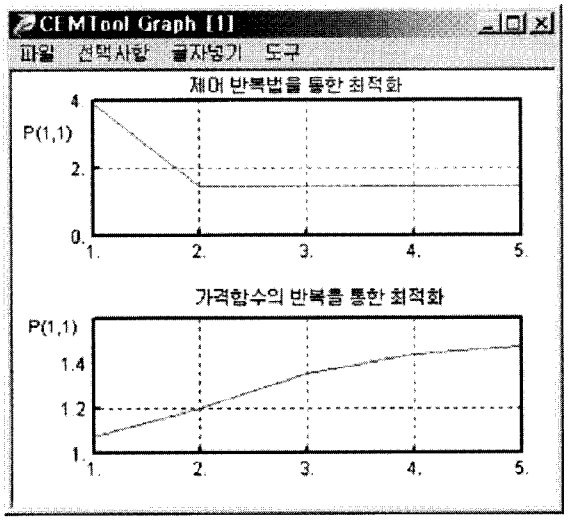

그림 2. 반복법을 통한 최적 해의 수치 계산

그 값들은 점차적으로 최적의 K 와 P 에 수렴한다. 지금까지 설명한 제어 반복법을 사용한 최적 제어는 아래의 시뮬레이션 코드로 CEMTool에서 수행할 수 있다.

시뮬레이션 코드

```
제어 반복법

A=[0.1741  0.3654  0.9823
   0.2569  0.8505  0.8556
   0.1049  0.9295  0.7692];
B=[0.8589 ; 0.2821 ;0.0916];
Q=eye(3); R=1 ;

iter=5 ;
K=place(A,B,[0.3 0.4 0.5]);
K=-K;
P=eye(3);
P_one=zeros(1,iter);
for k=1 : iter
   %P=Q+K'*R*K+(A+B*K)'*P*(A+B*K)
   P=dlyap((A+B*K)',Q+K'*R*K)
   P_one(k)=P(1,1);
   K=-inv(R+B'*P*B)*B'*P*A
end
```

그림 2에 윗부분을 보면 제어 반복법은 최적의 해를 2번 반복만에 구할 수 있음을 알 수 있다. 대수 리카티 방정식을 풀은 최적 해 P 의 1행 1열의 원소(1.4941)로 2번 만에 거의 수렴하는 것을 볼 수 있다.

6.2 가격함수의 반복을 통한 최적 제어 구하기

(20)에서 $J_n(x_i) = x_i^T P^{(n)} x_i$ 라고 놓으면 해당하는 최적화 문제는 다음과 같이 주어진다.

$$x_i^T P^{(n+1)} x_i = \min_u x_i^T Q x_i + u_i^T R u_i + x_{i+1}^T P^{(n)} x_{i+1} \quad (32)$$

모든 x_i 에 대하여 성립해야 하므로 아래와 같은 $P^{(n)}$ 에 관한 점화식이 얻어진다.

$$P^{(n+1)} = A^T P^{(n)} A - A^T P^{(n)} B(R + B^T P^{(n)} B)^{-1} B^T P^{(n)} A + Q \quad (33)$$

위의 초기 값을 $P^{(0)} = 0$ 으로 놓고 재귀적으로 풀면 $P^{(n)}$는 대수 리카티 방정식의 해로 수렴한다. 사실 (33)은 차분 리카티 방정식과 정확하게 일치하는 것으로 몇 가지 조건만 갖추면 시간이 흐름에 따라 대수 리카티 방정식으로 수렴함이 알려져 있다.

가격함수 반복법을 적용하기 위해 상태 방정식 모델 (25)를 다시 사용하자. 가격함수 반복법을 적용하는 시뮬레이션은 아래의 코드로 수행된다.

시뮬레이션 코드

```
가격함수 반복법

iter = 5 ;
P = eye(3);
P_one = zeros(1,iter);
for k=1 : iter
   K = -inv(R+B'*P*B) * B' * P * A ;
   P = Q + K'*R*K + (A+B*K)'*P*(A+B*K) ;
   P_one(k) = P(1,1) ;
end
```

그림 2의 아랫부분을 보면 가격함수 반복법에 의해 구한 해가 최적의 해에 점근적으로 수렴함을 알 수 있다. 전에 언급했듯이, 제어 반복법은 한 회당 계산량은 많으나, 짧은 반복 횟수(유한 상태 경우 유한 번의 횟수)내에 최적의 해에 가까운 결과를 얻을 수 있고, 가격함수 반복법은 한 회당 계산량은 적으나, 최적의 해에 수렴하는 속도가 늦다.

7. 결 론

본고에서는 확률 동적 자원 분배 문제에서 주로 많이 다루고 있는 마코프 의사 결정 과정 모델을 소개하고 최적으로 운용하기 위한 제어를 어떻게 결정하는지를 알아보았다. 상태 방정식에 이미 익숙한 제어 전공자를 위해서 이차형식 가격함수를 사용하여 최적 해를 구하는 과정을 설명하였다.

마코프 의사 결정 과정 모델은 주변의 많은 이산사건 시스템을 묘사할 수 있기 때문에 앞으로 많은 응용이 기대된다.

참 고 문 헌

[1] S. Haykin, *Neural Networks*, Prentice-Hall, 1999.

[2] F. L. Lewis, *Optimal control*, John Wiley & Sons, 1986.

[3] W. H. Kwon and S. Han, *Receding horizon control*, Springer, 2005.

[4] CEMTool : http://www.cemtool.co.kr

[5] MATLAB : http://www.mathworks.com

저 자 약 력

한 수 희

· 1974년 8월 26일생.
· 1998년 2월 서울대학교 전기공학부(공학사).
· 2000년 2월 서울대학교 전기 · 컴퓨터공학부(공학석사).
· 2003년 8월 서울대학교 전기 · 컴퓨터공학부(공학박사).
· 2002년 11월~2003년 8월 서울대학교 제어계측신기술연구센터 연구원.
· 2003년 9월~2005년 1월 서울대학교 BK 박사후 연구원.
· 2005년 1월~2005년 10월 스탠포드 방문연구원.
· 2005년 10월~2007년 4월 서울대학교 제어계측신기술연구센터 선임 연구원.
· 2007년 5월~현재 서울대학교 기계 · 항공공학부 BK 연구 교수.
· 관심분야 : 최적 제어 및 확률 제어, 이동 구간 제어, 시간 지연 시스템, 블록형 최적화, 추정 이론, 컴퓨터 이용 제어 시스템 설계.