

계층 최적화 기법에 의한 비선형 계통의 최적 제어에 관한 연구

전희영 · 박귀태 · 이종렬 · 이희정  
 고려대학교 공과대학 전기공학과

A study on the hierachical optimization methods  
 for the optimal control of nonlinear systems

Chun Hee Young: Park Gwi Tae: Lee Jong Ryeol:  
 Lee Hee Jeung  
 Dept. of Electrical Eng. Korea Univ.

ABSTRACT

In this paper, "Revised two-level costate prediction method" is developed to optimize the quadratic performance of a class of nonlinear dynamic systems. To show the merit of this algorithm, the proposed algorithm is compared with "The new prediction method" and "Two-level costate prediction method". Advantages of this algorithm are illustrated by applying it to three examples, turbine generator system, fermentation process, power control system in nuclear reactor.

1. 서론

최적 제어 이론을 적용하여 비선형 계통을 제어하려면, 높은 차수의 비선형 미분 방정식을 반복적으로 계산해야 하므로 많은 계산 시간 및 기억 용량이 요구된다. 특히 계통이 대규격인 경우에는, 최적해를 구하는데 필요한 컴퓨터 용량이 범용의 컴퓨터 용량을 초과하기 때문에 최적해를 구할 수 없다. 이러한 단점을 피하기 위한, 효과적인 제어 기법으로 등장한 것이 계층 최적화(Hierarchical optimization) 기법이다.

계층 최적화 기법이란 대규모 계통의 최적 제어 문제를 여러개의 부분계통(Subsystem) 문제로 나누어, 각 부분계통들은 자체의 국부적인 목표를 설정하고 이를 수행하는데, 이것이 전체 계통의 목표 관점에서 보아 합당한가 그렇지 아니한가를 상위 제어가 판단하여 새로운 지시(Coordination)를 내리는 방식이다.

본 논문은 계층 최적화 기법의 일종인 "개선된 2중 계층 예측 기법"(Revised two-level costate prediction method; RTLCPM)을 제시하고, 이 기법(2)의 효과적임을 보이기 위해 기존의 알고리즘인 "새로운 예측 기법"(The new prediction method; NPM), "2중 계층 부상태 예측 기법"(Two-level costate prediction method; TLCPPM)과 비교하였으며, 수치예로서 타빈 발전기 계통, 원자로 출력 제어 계통, 6-APA (6-aminopenicillanic acid) 생산 계통에 3가지 알고리즘을 적용하여 비교함으로써, 본 논문이 제시한 알고리즘의 타당성과 유효성을 시뮬레이션을 통해 보였다.

2. 개선된 2중 계층 부상태 예측 기법

다음과 같은 상태 방정식으로 표현되는 비선형 미선지 계통의 최적 제어 문제를 생각하자.

$$X(k+1) = f(X, U, k) \quad (2.1)$$

여기서 X는 n차의 상태 벡터이고, U는 m차의 제어 벡터이다. 최소화 할 평가지표는 다음과 같다.

$$J = \sum_{k=0}^{N-1} \|X(k) - X_d\|_Q^2 + \|U(k) - U_d\|_R^2 \quad (2.2)$$

여기서 X<sub>d</sub>, U<sub>d</sub>는 상태 벡터 및 제어 벡터들이 따라가야 할 목표치(Desired value)이고, Q, R은 각각 상태 벡터 제어 벡터들의 허용 행렬(Weighting matrix)이다.

식(2.1)의 상태 방정식은 항상 다음과 같은 형태로 정리 할 수 있다.

$$X(k+1) = A(X, U, k) + B(X, U, k) + D(X, U, k) \quad (2.3)$$

여기서 A와 B는 각각 X와 U의 블록 대각 요소를 가지는 행이고, D는 블록 비대각 요소를 가지는 행이다.

블록 비대각 요소와 각 기동량 요소들을 고정(Fix) 시키기 위해 예측 벡터 X, U를 도입하여 식(2.3)을 다시 쓰면

$$X(k+1) = A(X, X^*, U, k) + B(U, X^*, U, k) + D(X, U, k) \quad (2.4)$$

식(2.2), (2.4)로 주어지는 최소화 문제를 풀기위해 Hamiltonian을 정의하면

$$H = \frac{1}{2} \|X(k) - X_d\|_Q^2 + \frac{1}{2} \|U(k) - U_d\|_R^2 + \alpha(k+1) (A(X, X^*, U, k) + B(U, X^*, U, k) + D(X, U, k) - X(k) - X^*(k)) + T(k) (U(k) - U^*(k)) \quad (2.5)$$

여기서 β, T는 각각 X와 U를 X\*, U로 예측하기 때문에 생긴 부상태 변수(Costate variable)이다. 식(2.5)는 X, U가 미리 예측 됨으로써 다음과 같은 N개의 Hamiltonian로 분리 될 수 있다.

$$H = \sum_{i=1}^N H_i \\ H_i = \frac{1}{2} \|X_i(k) - X_{id}\|_{Q_i}^2 + \frac{1}{2} \|U_i(k) - U_{id}\|_{R_i}^2 + \alpha_i(k+1) (A_i(X_i, X_i^*, U_i, k) + B_i(U_i, X_i^*, U_i, k) + D_i(X_i^*, U_i, k) + \beta_i(k) (X_i(k) - X_i^*(k)) + T_i(k) (U_i(k) - U_i^*(k))) \quad (2.6)$$

식(2.6)의 Hamiltonian으로 부터 최적화 필요 조건을 구해보면

$$\frac{\partial H_i}{\partial U_i(k)} = 0 \text{ 로 부터} \\ R_i(U_i(k) - U_{id}) + (\frac{\partial B_i}{\partial U_i}(X_i^*, U_i^*, k) / \frac{\partial B_i}{\partial U_i}(X_i^*, U_i^*, k)) \alpha_i(k+1) + T_i(k) = 0 \quad (2.7)$$

$$\frac{\partial H_i}{\partial \alpha_i(k+1)} = X_i(k+1) \text{ 로 부터} \\ X_i(k+1) = A_i(X_i, X_i^*, U_i, k) + B_i(U_i, X_i^*, U_i, k) + D_i(X_i^*, U_i, k), X_i(0) = X_{i0} \quad (2.8)$$

$$\frac{\partial H_i}{\partial X_i(k)} = \alpha_i(k) \text{ 로 부터} \\ \alpha_i(k) = Q_i(X_i(k) - X_{id}) + (\frac{\partial A_i}{\partial X_i}(X_i, X_i^*, U_i, k) / \frac{\partial A_i}{\partial X_i}(X_i, X_i^*, U_i, k)) \alpha_i(k+1) + \beta_i(k), \alpha_i(k_f) = 0 \quad (2.9)$$

$$\frac{\partial H}{\partial X}(k) = 0 \text{ 로 부터}$$

$$X^*(k) = X(k) \quad (2.10)$$

$$\frac{\partial H}{\partial U}(k) = 0 \text{ 로 부터}$$

$$U^*(k) = U(k) \quad (2.11)$$

$$\frac{\partial H}{\partial \alpha}(k) = 0 \text{ 로 부터}$$

$$\beta(k) = (\frac{\partial (A(X^*, U^*, k) + B(U^*, X^*, k) + D(X^*, U^*, k))}{\partial \alpha}(k)) \alpha(k+1) \quad (2.12)$$

$$\frac{\partial H}{\partial \Gamma}(k) = 0 \text{ 로 부터}$$

$$\Gamma(k) = (\frac{\partial (A(X^*, U^*, k) + B(U^*, X^*, k) + D(X^*, U^*, k))}{\partial \Gamma}(k)) \alpha(k+1) \quad (2.13)$$

이상의 최적화 필요 조건을 만족시키기 위한 2중 구조 알고리즘은 다음과 같이 정리 할 수 있다.

1. 벡터  $\pi = (X^*, U^*, \alpha)$ ,  $U$ 의 궤적을 예측하고 반복 지표  $L$ 을 0으로 놓는다.
2.  $X^*, U^*, U$ 를 식(2.8)에 대입하여 새로운 상태 궤적을 구하고, 이 값들을 식(2.12)에 대입하여 부상태  $\beta$ 의 궤적을 구한다음, 식(2.9), (2.13)으로 부터 부상태  $\alpha, \Gamma$ 의 새로운 궤적을 구한다.

3. 만약  $X(k) = X^*(k) (k=1 \dots k_f)$ 이면 반복 개산을 중단하고 그 때의 입력 궤적을 최적 입력으로 기록한다. 그렇지 않으면 다음 식으로 각각의 궤적들을 수정하고, 반복 지표를 하나 증가시켜 2단계로 되돌아 간다.

$$\pi = \sigma \pi^{old} + (1-\sigma) \pi^*$$

여기서  $\sigma$ 은  $0 < \sigma < 1$ 인 유연 요소(Smoothing term)

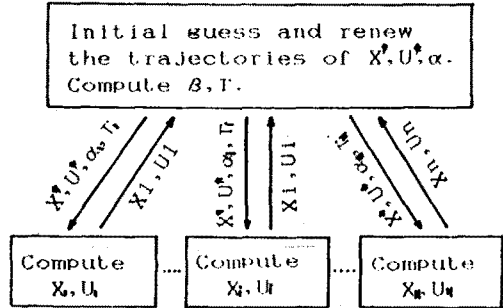
$$U^{**} = U^* - \tau (\frac{\partial H}{\partial U})$$

여기서  $\tau$ 는  $\tau > 0$ 인 스텝 사이즈(Step size)

이것을 계층 구조 블록 선도로 나타낸 그림(2.1)과 같다.

그림(2.1)에서 보는 바와 같이, 상위 제어기는  $X^*, U^*, \alpha$ 의 궤적을 예측 수정하고 부상태 변수  $\beta, \Gamma$ 를 계산하며, 하위 제어기는 상태와 입력의 궤적을 계산한다. 이와 같이 2중 구조 알고리즘으로써 최적해를 구하면 단일 컴퓨터로 계산한다 할지라도 준선형화(Quasi-linearization) 기법과 같은 단일 구조 제어 기법에 비해 계산 시간 및 기의 용량이 각각된다. 또한

이 기법은 2중 구조 알고리즘이므로 멀티프로세서를 이용하여 최적해를 구할 수 있다. 이 경우는 전체 계산 용량 및 기의 용량을 여러개의 프로세서들이 나누어 가질 수 있기 때문에, 이 알고리즘은 대규모 계층의 최적 제어 문제에도 적합하다.



Fig(2.1) Two-level block diagram of RTLCPM

3. 3가지 알고리즘의 비교 분석  
 새로운 예측 기법은, 비선형 방정식을 계층의 평형점에 대해 테일러 급수로 전개하여 2차 이상의 양의 상위 제어기에서 예측하여 고정(Fix)시킴으로써, 주어진 계층은 비선형이지만 하위 제어기는 선형 2차형식 부계층 문제(Linear quadratic subproblem)를 품고 상위 제어기는 하위 제어기에서 올라온 정보를 가지고 조정 변수(Coordination variable)를 계산하는 기법이다. 2중 계층 부상태 예측 기법은 2층에

Table(3.1) Comparison of three methods in the sense of algorithm

Algorithm Content of comparison	NIM	TLCPM	RTLCPM
Problem	$\text{Min } J = \frac{1}{2} \sum_{k=1}^{k_f} \ X(k) - X_d\ _Q^2 + \ U(k) - U_d\ _R^2$ subject to $X(k+1) = f(X, U, k)$		
Decomposition form of system equation	$X(k+1) = AX(k) + BU(k) + D(X^*, U^*, k)$	$X(k+1) = A(X^*, U^*, k)X(k) + B(X^*, U^*, k)U(k) + D(X^*, U^*, k)$	$X(k+1) = A(X, X^*, U^*, k) + B(U, X^*, U^*, k) + D(X^*, U^*, k)$
Initial guess trajectories	$X^*, U^*, \beta, \Gamma$	$X^*, U^*, \alpha, \Gamma$	$X^*, U^*, \alpha, U$
Determination of input trajectories	$\frac{\partial H}{\partial U} = 0$	$\frac{\partial H}{\partial U} = 0$	$U^{**} = U - \tau * (\frac{\partial H}{\partial U})$
Computation load of high-level controller	Initial guess and renew the trajectories of $X^*, U^*, \beta, \Gamma$	Initial guess and renew the trajectories of $X^*, U^*, \alpha, \Gamma$ Compute $\beta$	Initial guess and renew the trajectories of $X^*, U^*, \alpha$ Compute $\beta, \Gamma$
Computation load of low-level controller	Compute Riccati equation and $X_i, U_i$	Compute $X_i, U_i$	Compute $X_i, U_i$

Table(3.2) Comparison of three methods in the sense of feature

Algorithm contents of comparison	NPM	TLCPM	RTLCPM
Necessity of equilibrium point	Always required for the nonlinear system.	Required for the nonlinear system which $U(k)$ is not expressed of $X(k)$ , $\alpha(k)$ .	Not required for any system.
Storage requirement	Trajectories of $n(n/N+1)/2N+5n+3m$	Trajectories of $4(n+m)$ .	Trajectories of $4(n+m)$
Computation required	Integration of $n+n(n/N+1)/2N$ difference equations. Calculation of $n+m$ trajectories. Inversion of $N(n/N)*n/N$ matrix.	Integration of $2n$ difference equation. Calculation of $2(n+m)$ trajectories.	Integration of $2n$ difference equation. Calculation of $2(n+m)$ trajectories.
Convergence	On the case that the initial state value is far from the desired value, input constraints are severe, the initial guess trajectories are poor this algorithm diverges	On the case that input constraints are severe, the initial guess trajectories are poor this algorithm diverges.	With proper value of $\tau$ , this algorithm converges very well

서 소개한 RTLCPM의 알고리즘과 유사한 입력U를 최적화 필요 조건에 의해서 결정한다는 점이 다르다. 그러나 이 방법들은 다음과 같은 단점들이 있다.

1. 제어하고자 하는 계통의 입력에 심한 구속 조건이 있으면 이 알고리즘들은 수렴하지 않는다.
2. 알고리즘의 수렴성이 초기 예측 값에 민감하다. 따라서 초기 예측 값이 최적 값에서 너무 떨어져 있으면 이 알고리즘들은 발산한다.
3. NPM을 적용하여 비선형 계통을 제어하려면 항상 계통의 평형 점을 미리 알아야 하며, 최적화 필요 조건을 적용하여 입력에 대한 식을 유도했을 때 이 식이 상태와 부상태만의 함수로 표현되지 않는 계통에 대하여 TLCPM를 적용하려면 NPM에 마찬가지로 계통의 평형점을 미리 알아야 한다.

본 논문에서 제시한 RTLCPM은 이러한 단점들을 해결하기 위해 제어 입력을 경사 기법(Gradient method)으로써 결정한다. 경사 기법은 수렴성이 좋고 초기 예측 값에 민감하지 않다는 장점이 있는 반면에, 최적 상태에 가까워 가면 수렴 속도가 극히 느리다는 단점이 있으나 본 논문이 제시한 알고리즘에서는 입력 값의 뿐만 아니라 상태 값과 부상태 값도 상위 제어기에서 수정하기 때문에 이러한 난제도 피할 수 있다.

표(3.1)은 3가지 기법을 알고리즘 면에서 비교한 것이다. 우선 초기예측이 필요한 과정을 보면 NPM과 TLCPM에서는 부상태 T의 값을 예측하는데 반해, RTLCPM에서는 입력 U의 값을 예측한다. 부상태의 값을 정확히 예측한다는 것은 대단히 어려운 일이나, 입력 값은 계통의 물리적인 의미로 부터 최적 값에 가깝게 선정할 수 있다. 따라서 RTLCPM는 다른 알고리즘에 비해 수렴 속도가 빠르다.

입력의 결정 방법면에서 보면 LPM과 TLCPM은 최적 필요조

건으로 결정하는데 반해 RTLCPM에서는 경사기법으로 결정한다. 입력에 심한 구속조건이 있는 계통을 제어할때 최적필요조건으로 입력을 결정하면 입력의 구속조건으로 인해 최적조건이 깨어지지만 경사기법으로 결정하면 입력의 구속조건을 반복 계산때마다 조금씩 수정하여 결국 최적값으로 가게 하기 때문에 RTLCPM은 입력에 심한 구속조건이 있는 계통에도 적용될 수 있다.

하위 제어기의 계산량을 비교해보면 NPM은 선형 2차 형식 제어해를 얻기 위한 Riccati 방정식을 풀어야 하므로 다른 기법에 비해 하위 제어기의 계산분담이 크다.

표(3.2)는 3가지 기법을 특징면에서 비교한 것이다.

기억용량이나 계산용량면에서 보면 TLCPM과 RTLCPM는 거의 같지만 NPM는 다른 알고리즘에 비해 훨씬 큰 용량이 필요하다. 이 이유는 NPM에서는 2점 경계치 문제(Two-point boundary value problem)를 해결하기 위한 Riccati 방정식을 풀기 때문이다. 수렴성이나 평형점의 필요유무 면에서도 보면 RTLCPM가 다른 기법들에 비해 보다 더 광범위하게 적용될 수 있음을 알 수 있다.

4. 시뮬레이션 및 검토

본 장에서는 2장에서 제시한 RTLCPM의 타당성과 유효성을 밝히기 위하여, 3가지 기법을 3개의 비선형 계통에 적용하여 비교하기로 한다. 원래 NPM는 연속계 계통에 적용될 수 있도록 제안되었으나 본 논문에서 제시한 알고리즘과 비교하기 위해 이산시 알고리즘으로 변환시켜 시뮬레이션 하였다.

모든 시뮬레이션은 IBM1411의 단일 컴퓨터로 하였음을 밝혀 둔다.

4.1 터빈 발전기 계층 <6>

이 절에서는 3가지 기법이 모두 적용될 수 있는 일반적인 계층으로서, 여자기와 조속기가 있으며 무한대 모션 계층에 변압기와 송전선으로 연결된 단일 동기 발전기를 최적화 모델로 선정 하였다 여기서 취급되는 외란은 변압기 계층측에 3상 지력 사고가 일어나 얼마후 자연 복구되는 것으로 가정하였다. 이 계층의 상태 방정식은 다음과 같이 쓸 수 있다.

$$\begin{aligned} x_1(k+1) &= x_1(k) + \Delta T x_2(k) \\ x_2(k+1) &= (1 - \Delta T c_1) x_2(k) - \Delta T c_2 x_3(k) \sin x_4(k) - 0.5 \Delta T c_3 \sin 2x_4(k) + \Delta T x_6(k) / m \\ x_3(k+1) &= (1 - \Delta T c_4) x_3(k) + \Delta T x_4(k) + \Delta T c_5 \cos x_4(k) \\ x_4(k+1) &= (1 - \Delta T k_1) x_4(k) + \Delta T k_2 x_2(k) + \Delta T k_3 u_1(k) \\ x_5(k+1) &= (1 - \Delta T k_4) x_5(k) + \Delta T k_5 x_2(k) \\ x_6(k+1) &= (1 - \Delta T k_6) x_6(k) + \Delta T k_7 u_2(k) \end{aligned} \quad (4.1)$$

여기서

- $x_1$  : 부하각
- $x_2$  : 속도 변동(dx/dt)
- $x_3$  : 개자 자속 쇄고수
- $x_4$  : 터빈 유입 증기력
- $x_5$  : 기계적인 입력 토크
- $x_6$  : 여자기의 개자 전압
- $u_1$  : 여자기에 가해지는 입력 전압
- $u_2$  : 조속기의 속도 조절기여 조정값

각 계수들의 값은 다음과 같다.

$$c_1=2.165 \quad c_2=13.957 \quad c_3=-55.565 \quad c_4=1.02$$

$$c_5=4.0491 \quad k_1=9.4425 \quad k_2=1.019 \quad k_3=5.$$

$$k_4=2.04 \quad k_5=2.04 \quad k_6=1.5 \quad k_7=0.5 \quad m=0.0338$$

사고가 제압된 후 빠른 시간내에 계층의 안정도를 높이고 과도 진동을 없애는 것이 최적화 문제의 목적이므로, 평가지표를 다음과 같이 설정한다.

$$J = \sum_{k=1}^N (Q_1 x_1(k) - x_{1d})^2 + (Q_2 x_2(k) - x_{2d})^2 + R_1 u_1(k) - u_{1d})^2 + R_2 u_2(k) - u_{2d})^2 \quad (4.2)$$

4.2 6-아미노 페니실라민 산(6-aminopenicillanic acid)을 생산하기 위한 발효 계층 <6>

본 절에서 연구 대상으로 정한 계층은 양성이 사미 산에 변수와 부실태 변수만으로는 표현되지 않는 계층으로서, 효소를 이용하여 페니실린 G(Benzylpenicillin)을 개수분해시켜, 6-APA를 만들어 내는 공생이다. 계층의 상태 방정식은 다음과 같이 표현된다.

$$\begin{aligned} x_1(k+1) &= (1 - \Delta T k_1) x_1(k) - \Delta T (k m x_1(k) x_3(k) / (k s + x_1(k))) \\ x_2(k+1) &= (1 - \Delta T k_2) x_2(k) + \Delta T (k m x_1(k) x_3(k) / (k s + x_1(k))) \\ x_3(k+1) &= (1 - \Delta T k) x_3(k) \end{aligned} \quad (4.3)$$

여기서

$$k s = 1.503 \text{ (mg/ml)}$$

$$k m = 668.76 (1 - (v_2 - 8.5)^2 / 17.4) \exp(2.87 * 10^3 (3.155 * 10^{-3} v_1))$$

$$k 3 = \exp(-7.4 - 1.07 * 10^4 (v_1 - 3 * 10^3) + 2.05 (v_2 - 9.5))$$

$$k 2 = \exp(-1.54 - 1.243 * 10^4 (v_1 - 3 * 10^3) + 2.05 (v_2 - 9.5))$$

$$k = 2.06 * 10^4 \exp(-3.655 * 10^4 v_1)$$

- $x_1$  : 페니실린 G의 농도
- $x_2$  : 6-APA의 농도
- $x_3$  : 효소의 농도
- km: 효소의 최대 반응비
- k3: 페니실린 G의 감소비
- k2: 6-APA의 감소비

$k_1$ : 효소 활동의 역제비

$k_s$ : Michaelis-Menten 포화 상수

$v_1$ : 절대 온도의 역수 ( $v_1 = 1 / (u_1(k) + 273.2)$ )

$v_2$ : pH값 ( $v_2 = u_2(k)$ )

제어 입력은 물리적인 제한 조건이 있기 때문에 다음과 같은 구속 조건을 가진다.

$$u_1(k) < 44 \text{ } ^\circ\text{C}$$

$$6.5 < u_2(k) < 10 \quad (4.4)$$

여기서  $u_1$ 은 섭씨 온도이고,  $u_2$ 는 pH값이다.

제어 목적은 계층이 정상 상태에서 운전중 미지의 외란으로 인하여 상태값이 변화했을 경우, 수시간(6시간) 이내에 다시 정상 상태로 가개끔 하는 것이다. 따라서 평가지표를 다음과 같이 설정 한다.

$$J = \frac{1}{2} \sum_{k=1}^N (\|X(k) - X_d\|_{Q_1}^2 + \|U(k) - U_d\|_{R_1}^2) \quad (4.5)$$

4.3 원자로 출력 제어 계층 <7>

이 절에서는 입력이 심한 구속 조건을 가지는 계층의 예로서, 원자로 출력 제어 계층에 적용시켜 본다. 즉 원자로의 출력 요구가 어느 준위에서 다른 준위로 변할 때, 이에 대응하는 원자로 출력 제어 계층을 고려한다. 제어 대상 계층은 다음과 같은 상태 방정식으로 표현 된다.

$$\begin{aligned} x_1(k+1) &= (1 - 7.3 \Delta T) x_1(k) + 7.3 \Delta T x_2(k) + \Delta T x_3(k) u(k) - \Delta T x_4(k) x_6(k) \\ x_2(k+1) &= (1 - 0.0787 \Delta T) x_2(k) + 0.0787 \Delta T x_1(k) \\ x_3(k+1) &= (1 - 0.1 \Delta T) x_3(k) + 0.1 \Delta T x_4(k) \end{aligned} \quad (4.6)$$

여기서

- $x_1$  : 중성자 밀도(원자로 출력)와 그 초기치와의 비
- $x_2$  : 지연 중성자의 선형화 농도에 대응하는 출력과 그 초기치와의 비
- $x_3$  : 평균 핵연료 온도와 그 초기치와의 비
- $u$  : 반응도와 그 초기치와의 비

반응도와 그 변화율은 물리적인 제한 조건이 있기 때문에, 제어 변수  $u$ 는 다음과 같은 구속조건을 가진다.

$$1 < u(k) < 4$$

$$-0.2 \Delta T < u(k+1) - u(k) < 0.2 \Delta T \quad (4.7)$$

원자로가 초기 상태에서 운전중 최종 시간 100초 이내에 30%의 출력이 요구되면는 가장하여, 평가지표는 다음과 같이 설정 한다.

$$J = \frac{1}{2} \sum_{k=1}^N (\|X(k) - X_d\|_{Q_1}^2 + \|U(k) - U_d\|_{R_1}^2) \quad (4.8)$$

4.4 결과 및 검토

표 (4.1)은 3가지 알고리즘을 3가지 계층에 적용시킨 결과를 비교한 것이다.

우선 반복횟수 면에서 보면 NPM가 가장 적으나 계산시간 및 기억 용량 면에서 보면 다른 알고리즘에 비해 월등히 떨어진다. 이것은 NPM의 알고리즘이 2점 경계치 문제를 풀기위한 Riccati 해를 구하기 때문이다. TLCPM 과 RTLCPM을 비교해 보면 기억용량 면에서는 같지만 RTLCPM의 수렴속도가 훨씬빠르기 때문에 계산 시간도 차이를 알 수 있다. 원자로 출력 제어 계층에 적용시킨 결과를 보면, RTLCPM은 반복 횟수 65 일때 평가가 0.2\*10E-2로서 좋은 수렴성이 있으나 다른 알고리즘들은 입력의 심한 구속조건 때문에 전혀 수렴하지 않는다. 최종 평가 지표 값은 각 알고리즘에 의한 최적 제어 기법의 값이 같기 때문에 비슷 하다.

5. 결론

본 논문에서는 비선형 계통의 최적 제어를 위해 계층 최적화 기법인 "개선된 2중 계층 부상태 예측 기법"을 제시하였고 이 기법이 효과적임을 선보이기 위해 다른 2가지의 알고리즘("새로운 예측 기법", "2중 계층 부상태 예측 기법")과 비교하였다.

이 3가지 알고리즘을 3개의 계통(터빈 발전기 계통, 6-APA 생산계통, 원자로 출력 제어계통)에 적용하여 시뮬레이션 결과 다음과 같은 결론을 얻었다.

1. 3가지 알고리즘을 기억 용량면에서 비교해보면 TLCPM와 RTLCPM는 같고, NPM는 이들 보다 훨씬 더 많이 요구된다  
2. 각 반복 계산 때마다의 계산 용량을 비교해 보면, TLCPM와 RTLCPM는 같고 NPM는 Riccati해법 구하기 때문에 전자보다 훨씬 많다.

3. 입력에 심한 구속 조건이 있는 계통을 제어할 경우, NPM와 TLCPM는 수렴하지 않으나 RTLCPM는 좋은 수렴성이 있다.

4. 비선형 계통의 최적 제어 문제에 NPM를 적용시키려면 미리 계통의 평형점을 알아야 하며, 입력의 표현식이 상태 변수와 부상태 변수만의 함수가 아닌 계통에 TLCPM를 적용시키려면 계통의 평형점을 알아야만 하나, RTLCPM에서는 어느 계통에 대해서도 알 필요가 없다.

5. 계산 시간면에서 보면 RTLCPM가 다른 알고리즘들보다 수렴성이 좋기 때문에 계산 시간이 가장 적다.

결론적으로 말하면, 본 논문이 제시한 RTLCPM는 비선형 계통을 최적 제어하는 데 있어서, 수렴성도 빠르고 기억 용량도 적게 요구되며 계통의 특성에 관계없이 광범위하게 사용될 수 있다.

본 논문에서는 비선형 계통의 최적제어 문제만 다루었으나, 대규모 선형 계통의 최적제어 문제에도 RTLCPM를 적용시키면 좋은 결과를 얻을 수 있으리라고 생각된다.

\* 참고 문헌

1. D.E.Kirk; "Optimal control theory", Prentice-Hall, Inc., 1977
2. M.G.Singh; "Dynamical hierarchical control", North-Holland, Amsterdam, 1977
3. M.Hassan, M.G.Singh; "The optimization of nonlinear systems using a new two-level method", Automatica. Vol.12, 359-362, 1976
4. M.G.Singh and M.Hassan; "A two-level costate prediction algorithm for nonlinear systems", Automatica 13 95-96, 1977
5. B.K.Mukhopadhy and O.P.Malik; "Solution of nonlinear optimization problem in power systems", IJC 17, 10 41-1058, 1973
6. A.S.Fawzy, O.R.Hinton and M.S.Mahmoud; "Hierarchical techniques for on-line microprocessor control of a fermentation process", INT.J.Systems. SCI, Vol.14, No.1, 19-29, 1983
7. K.Oguri and Y.Ebizuca; "Synthesis of digital control systems for nuclear reactors: Part1 Optimal solutions to power change control systems" J.N.S.&T., Vol.12, No.7, 391-401, 1975

Table(4.1) Simulation results

	Turbine generator system			Fermentation process			Power control system		
	NPM	TLCPM	RTLCPM	NPM	TLCPM	RTLCPM	NPM	TLCPM	RTLCPM
Initial value of state and control variable	$X_0 = (0.7105 \ 0. \ 4.2 \ 0.8 \ 0.8 \ 0.5)$			$X_0 = (2. \ 7. \ 0.01)$			$X_0 = (1. \ 1. \ 1.)$ $U_0 = (1.)$		
Desired value of state and control variable	$x_d = 0. \ x_{d4} = 3.969$ $u_d = 0. \ u_{d4} = 0.$			$X_d = (0. \ 8.98 \ 0.009)$ $U_d = (40. \ 7.5)$			$X_d = (3. \ 3. \ 3.)$ $U_d = (3.)$		
Weighting matrix	Q11=0.4 Q33=0.4 R11=10. R22=10.			Q=diag(1. 1. 1.) R=diag(100. 10.)			Q=diag(5*10 5*10 25.) R=2.		
Equilibrium point of system	$X_e = (0 \ 0 \ 3.969 \ 0 \ 0 \ 0)$ $U_e = (0 \ 0)$			$X_e = (0. \ 8.98 \ 0.009)$ $U_e = (40. \ 7.5)$			$X_e = (3. \ 3. \ 3.)$ $U_e = 3.$		
Termination criterion Error	0.01			0.01			5.*10		
Subsystem of NPM algorithm	$X1 = (x_1 \ x_2 \ x_3)$ $X2 = (x_4 \ x_5 \ x_6)$			$X1 = (x_1)$ $X2 = (x_2 \ x_3)$			$X1 = (x_1)$ $X2 = (x_2 \ x_3)$		
Soothing term $\sigma$	0.3			1.			0.9		
Step size $\tau$	0.07			0.1			0.1		
Sampling time $\Delta T$	0.05(sec)			0.05(18min/step)			0.3(sec)		
Iteration number	18	41	26	5	7	5	Diverge		65
Computation time (sec)	8	3	2	7	2	1.5			12
Storage requirement (k bite)	1.8	1.28	1.28	4	3	3			10
Error	2.5*10 <sup>3</sup> 9*10 <sup>4</sup> 4.3*10 <sup>4</sup>			4.3*10 <sup>4</sup> 4.8*10 <sup>8</sup> 2*10 <sup>8</sup>			0.2*10		
J	0.738	0.732	0.74	2.12	2.12	2.13			10*10

Error =  $\sum_{k=0}^{\infty} x_i(k) - x_i^*(k)$

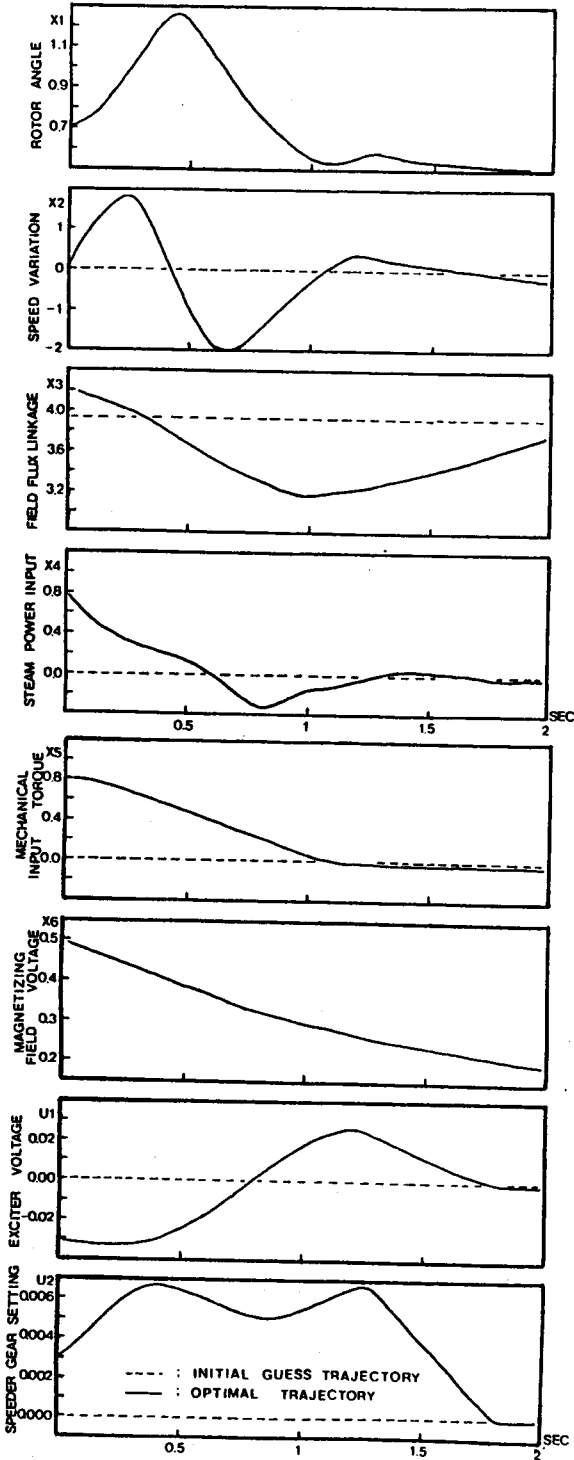


Fig. (4.1) Optimal trajectories of turbine generator system

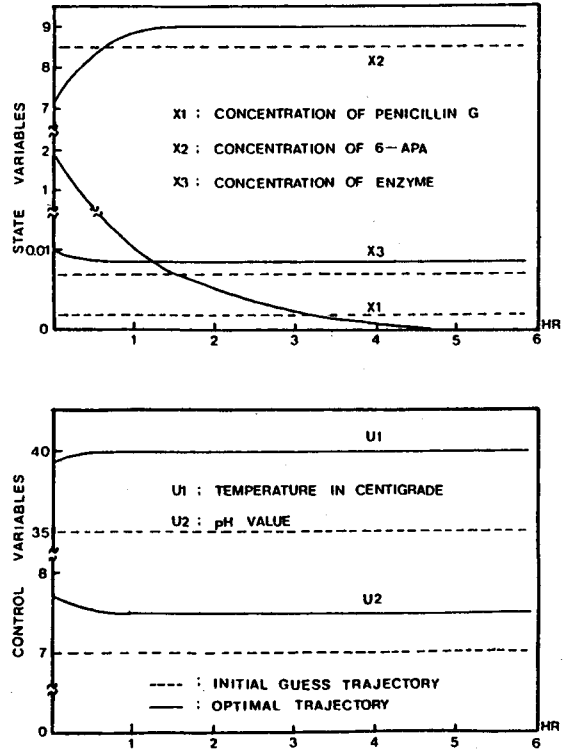


Fig. (4.2) Optimal trajectories of fermentation process

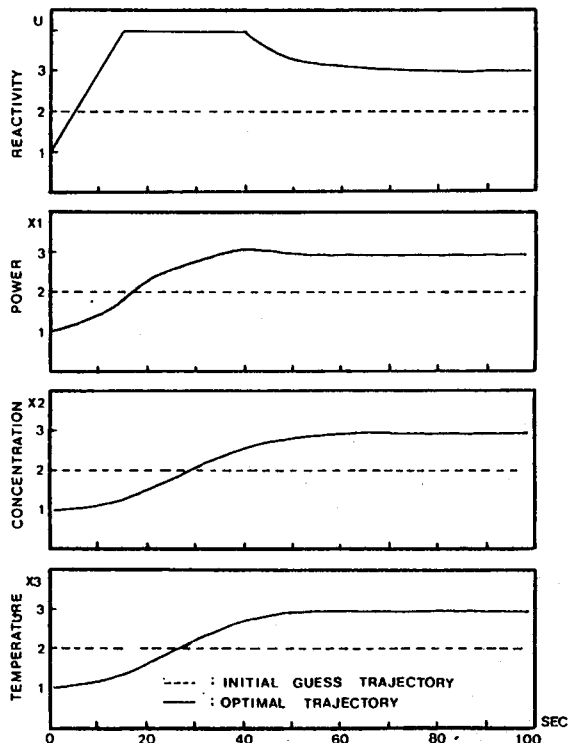


Fig. (4.3) Optimal trajectories of nuclear power plant system