

論文99-36T-6-4

대규모 비선형 시스템의 새로운 계층별 최적제어

(A New Approach for Hierarchical Optimization of Large Scale Non-linear Systems)

朴俊勳*, 金鐘夫**

(Joon-Hoon Park and Jong-Boo Kim)

요 약

대규모 시스템 제어와 관련하여 몇 개의 부시스템으로 분할하여 처리하는 계층별 제어이론이 많이 연구되어 왔으며 특히 차수가 높은 대규모 비선형 시스템의 경우에 고차의 비선형 미분방정식을 동시에 적분을 해야하고 많은 계산량을 필요로 하므로 해를 구하기가 어려운 문제가 있다. 1980년대 Singh과 Hassan은 상호예측 알고리즘(two level prediction algorithm)을 제시한바 있고 이 방법은 비선형 대규모 시스템의 최적제어에 효과적이나 시스템 행렬 Q, R, S, H 에 따라 제한된 최적화 구간에서만 성립되는 등 최적화 구간의 길이와 수렴성 여부가 행렬 값에 영향을 받는 알고리즘상의 단점이 있다. 본 연구에서는 수렴조건으로부터 평가함수에 구속조건(quadratic penalty term)을 부여하지 않는 새로운 개선된 알고리즘을 제시 적용하여 시스템 행렬 결정을 위한 과정 없이 수렴속도의 향상과 함께 최적의 수렴성 및 최적화 구간을 얻도록 했다. 분할된 비선형 시스템의 최적제어를 위해서는, 대수 반복연산만으로 2점 경계치 문제(two point boundary value problem)를 해결함으로써 기존의 수치 해석법에 비해 연산이 간단한 블록펄스 변환 방법을 사용해서 처리했다.

Abstract

This paper presents a new possibility of calculating optimal control for large scale which consist of non-linear dynamic sub-systems using two level hierarchical structures method. And the proposed method is based on the idea of block pulse transformation to simplify the algorithm and its calculation. This algorithm used an expansion around the equilibrium point of the system to fix the second and higher order terms. These terms are compensated for iteratively at the second level by providing a prediction for the states and controls which form of a part of the higher order terms. In this new approach the quadratic penalty terms are not used in the cost function. This allows convergence over a longer time horizon and also provides faster convergence. And the method is applied to the problem of optimization of the synchronous machine. Results show that the new approach is superior to conventional numerical method or other previous algorithms.

I. 서론

* 正會員, 忠州大學校 制御計測工學科

(Dept. of Control & Instrumentation Eng., Chungju Nat'l University)

** 正會員, 仁德大學 메카트로닉스과

(Dept. of Mechatronics, Induk College)

接受日字:1999年4月15日, 수정완료일:1999年6月18日

최근 로봇이나 항공기와 같이 첨단기술을 이용한 시스템 활용이 증가하고 있고 이러한 시스템에 포함된 비선형계의 동작에는 정밀성, 안정성 등의 엄격한 제한요소가 요구되므로 이의 해결을 위한 비선형계의 제어이론에 대한 연구의 필요성이 요구되고 있다. 비선형계

의 최적제어 이론 중 일반적인 방법은 수치해석기법을 이용한 급경사법(steepest descent), 준선형화법(quasilinearization), 반복기법(recursive algorithm) 등이 있으나 이러한 방법들은 차수가 높은 대규모 시스템의 경우에 고차의 비선형 미분방정식을 동시 적분과 많은 계산량을 필요로 하므로 해를 구하기가 어려운 문제가 있다.^[1] 따라서, 대규모 시스템을 몇 개의 부시스템으로 분할하여 처리하는 비선형계의 계층별 제어이론이 많이 연구되어 왔으며 1980년대 Singh와 Hassan은 비선형 대규모 시스템을 2계층으로 분할한 후 계층간의 상호예측원리를 이용해서 조정 예측자(predictor) 변수를 조정하는 2계층 상호예측 알고리즘(two level prediction algorithm)을 제시했다.^[2] 이 방법은 선형 항과 비선형 항을 분리하여 비선형 항만을 조정 예측자에서 조정하는 형태이므로 비선형 대규모 시스템의 최적제어에 효과적이다. 그러나 Singh와 Hassan 등이 제시한 기존의 알고리즘은 시스템 행렬 Q, R, S, H 를 결정하기 위한 별도의 과정이 요구되며 제한적인 최적화 구간을 갖는 등 최적화 구간의 길이와 수렴성 여부가 영향을 받고 경우에 따라서는 수렴하지 못하는 커다란 단점을 지니고 있다.^[6] 본 연구에서는 수렴조건 k 의 고찰을 통해 평가함수에 구속조건(quadratic penalty term)을 부여하지 않는 새로운 개선된 알고리즘을 적용해서 시스템 행렬 결정을 위한 과정 없이도 최적의 수렴성을 얻도록 했으며 수렴속도의 저하 및 최적화 구간이 제한되는 문제점 등을 해결하였다. 분할된 비선형 시스템의 최적제어를 위해서 블록펄스 변환을 했으며 대수 반복연산만으로 2점 경계치 문제(two point boundary value problem)를 해결하기 때문에 기존의 수치 해석법에 비해 연산이 간단함을 고찰하였다.

II. 개선된 대규모 비선형계의 계층별 최적제어

1. 구속조건(quadratic penalty term)의 고려
다음의 비선형 시스템을 고려한다.

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (1)$$

여기서 $\mathbf{x}(t) \in \mathbb{R}^n$ 는 n 차 상태벡터, $\mathbf{u}(t) \in \mathbb{R}^m$ 는 m 차 제어벡터이며 일반적으로 \mathbf{f} 는 상태벡터와 제어벡터의

함수이다. 이때 최소화할 평가함수는 다음과 같다.

$$J = \frac{1}{2} \int_{t_0}^{t_f} (\|\mathbf{x}(t)\|_Q^2 + \|\mathbf{u}(t)\|_R^2) dt \quad (2)$$

식(1)의 비선형 시스템의 동작점(equilibrium point)을 원점으로 가정하고, 테일러(Taylor)급수로 전개한 후 1차 이상인 항을 모두 무시하면 동작점에서 선형근사화가 이루어지고 식(3)과 같다.^[2]

$$\dot{\mathbf{x}}(t) = \mathbf{A}^* \mathbf{x}(t) + \mathbf{B}^* \mathbf{u}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) - \mathbf{A}^* \mathbf{x}(t) - \mathbf{B}^* \mathbf{u}(t) \quad (3)$$

여기서 \mathbf{A}^* , \mathbf{B}^* 는 선형화 행렬로 다음과 같이 정의한다.

$$\mathbf{A}^* = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_{\mathbf{x}(t), \mathbf{u}(t)=0}, \quad \mathbf{B}^* = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right]_{\mathbf{x}(t), \mathbf{u}(t)=0} \quad (4)$$

선형화된 \mathbf{A}^* , \mathbf{B}^* 행렬에서 대각(diagonal) 혹은 블록대각(block diagonal) 부분을 \mathbf{A} , \mathbf{B} 로 표현하고 비선형, 비대각 항을 $\mathbf{D}(\mathbf{x}, \mathbf{u}, t)$ 로 정의하면 식(3)은 다음과 같다.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{D}(\mathbf{x}, \mathbf{u}, t) \quad (5)$$

여기서 \mathbf{C}_1 , \mathbf{C}_2 는 행렬 \mathbf{A}^* , \mathbf{B}^* 의 비대각(off diagonal) 부분이다.

$$\mathbf{D}(\mathbf{x}, \mathbf{u}, t) = \mathbf{C}_1 \mathbf{x}(t) + \mathbf{C}_2 \mathbf{u}(t) + \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) - \mathbf{A}^* \mathbf{x}(t) - \mathbf{B}^* \mathbf{u}(t) \quad (6)$$

Singh와 Hassan은 제시한 알고리즘의 수렴성을 높이기 위하여 식(7), (8)의 구속조건(quadratic penalty term)을 평가함수에 추가함으로써 다음의 수정된 평가함수를 얻어 이를 최소화하는 방법을 취했다.

$$\mathbf{x}^o(t) = \mathbf{x}(t) \quad (7)$$

$$\mathbf{u}^o(t) = \mathbf{u}(t) \quad (8)$$

$$J = \frac{1}{2} \int_0^{t_f} \{ \|\mathbf{x}\|_Q^2 + \|\mathbf{u}\|_R^2 + \|\mathbf{x} - \mathbf{x}^o\|_S^2 + \|\mathbf{u} - \mathbf{u}^o\|_H^2 \} dt \quad (9)$$

여기서 \mathbf{S} , \mathbf{H} 는 블록 대각행렬이다. 이제 J 를 최소화시키는 문제를 해결하기 위해서 해밀토니안(Hamiltonian)의 정의로부터 먼저 상위계층의 조정 예측자를 구하면 식(7), (8)과 함께 식(10) 및 식(11)

을 얻는다.

$$\begin{aligned} \Pi(t) = & -S(\mathbf{x}(t) - \mathbf{x}^o(t)) \\ & + \left\{ C_1^T + \left[\frac{\partial f^T}{\partial \mathbf{x}(t)} \right]_{x=x^o, u=u^o} - A^{*T} \right\} \cdot \lambda(t) \end{aligned} \quad (10)$$

$$\begin{aligned} \beta(t) = & -H(\mathbf{u}(t) - \mathbf{u}^o(t)) \\ & + \left\{ C_2^T + \left[\frac{\partial f^T}{\partial \mathbf{u}(t)} \right]_{x=x^o, u=u^o} - B^{*T} \right\} \cdot \lambda(t) \end{aligned} \quad (11)$$

위 식에서 $\lambda(t) \in R^n$ 는 n차 상호상태벡터(costate vector)이며 $\Pi(t) \in R^n$ 는 n차 라그랑제 승수(Lagrange multipliers)이고, $\beta(t) \in R^m$ 는 m차 라그랑제 승수이다.

2. 수렴조건 $\|k\| < 1$ 의 고려

Singh과 Hassan은 제시한 알고리즘의 수렴조건이 시스템의 Q, R, S, H 행렬에 영향을 받음을 밝힌바 있다.^[3] 따라서 기존의 알고리즘에서는 수렴되는 최적화 구간을 증가시키고 수렴조건을 만족시키기 위해 별도의 시뮬레이션을 통한 Q, R, S, H 행렬의 적절한 조작이 필요하다.^[4] 실제로 기존의 알고리즘은 제한된 최적화 구간에서만 수렴을 하는 단점을 지니며 더욱이 시행착오법을 통한 행렬의 조작이 적절치 못할 경우 최적화 구간이 확장됨에 따라서 시스템이 수렴하지 않는 등 Q, R, S, H 행렬을 어떻게 설정하느냐에 따라 결과가 극히 불안정하게 나타나는 알고리즘상의 커다란 단점을 지니고 있다.^[11]

본 연구에서는 이러한 단점을 해결하고 개선된 알고리즘을 얻기 위해 수렴조건을 고려했다. 먼저 식(7)~(8) 및 식(10)~(11)로부터 수렴조건 판단을 위한 식(12)를 얻을 수 있는데 여기서 L은 상위계층 조정 예측자의 반복지표를 의미하고, L+1은 L의 다음 반복구간(iteration)을 의미하며 I 및 E는 식(13), (14)와 같다.^[5]

$$I^{L+1}(t) = kI^L(t) + E \begin{pmatrix} \mathbf{x}(t) \\ \lambda(t) \end{pmatrix} \quad (12)$$

$$I^{LT}(t) = [\mathbf{x}^{oL}(t), \Pi^L, \mathbf{u}^{oL}(t), \beta^L(t)] \quad (13)$$

$$E = \begin{pmatrix} I & & & 0 \\ -S & -A^{*T} & C_1^T + \left[\frac{\partial f^T}{\partial \mathbf{x}(t)} \right]_{x=x^o, u=u^o} & \\ 0 & & -R^{*-1} & B^T \\ 0 & -HR^{*-1} & B^T - B^{*T} + C_2^T \left[\frac{\partial f^T}{\partial \mathbf{u}(t)} \right]_{x=x^o, u=u^o} & \end{pmatrix} \quad (14)$$

또한 행렬의 최대 고유치(maximum eigenvalue)인 k 는 식(15)로 정의된다.

$$k = \begin{pmatrix} 0 & 0 & 0 & 0 \\ S & 0 & 0 & 0 \\ 0 & 0 & (R)^{*-1}H & -R^{*-1} \\ 0 & 0 & (H - HR^{*-1}H) & HR^{*-1} \end{pmatrix} \quad (15)$$

수렴조건과 관련하여 식(12)에서 L이 무한대로 접근함에 따라 I(t)는 영(zero)으로 수렴해야하며 이를 위해선 $\|k\| < 1$ 의 수렴조건을 만족해야만 한다. 따라서 $S=H=0$ 로 두면 k 의 최대 고유치인 $\|k\|$ 가 가장 작은 값을 지니며 더 이상의 S, H 행렬에 대한 조작이 필요 없게된다. 이 경우 수렴조건을 최적으로 만족시키므로 기존의 알고리즘에서 수렴조건의 고려 없이 시행착오법을 통해 이루어졌던 어떤 S, H의 조작 값보다 수렴조건을 만족과 함께 최적화 구간을 증가시킬 수 있으며 빠른 수렴속도를 가짐은 물론 최적화 구간에 관계없이 안정적인 수렴 결과를 얻을 수 있다.

이제 수렴조건으로부터 얻은 $S=H=0$ 로 두고 독립된 부시스템의 분할을 고려하면 식(6)의 비대각 항 D(x, u, t)는 식(7), (8)로 고정되었으므로 전체 시스템을 N개의 독립적인 부시스템으로 분할하여 다음과 같이 독립된 최적화 문제를 유도할 수 있다. 먼저 $u_i(t)$ 를 구하면 다음과 같다.

$$u_i(t) = R_i^{-1}[-B_i^T \lambda_i(t) - \beta_i(t)] \quad (16)$$

또한 $\dot{x}_i(t)$ 에 대해 정리하면 식(17)을 얻는다.

$$\begin{aligned} \dot{x}_i(t) = & A_i x_i(t) + B_i R_i^{-1}[-B_i^T \lambda_i(t) - \beta_i(t)] \\ & + D_i(\mathbf{x}_i^o(t), \mathbf{u}_i^o(t), t) \end{aligned} \quad (17)$$

$$x(t_0) = x_0 \quad (18)$$

상호상태 벡터 $\lambda_i(t)$ 는 식(19)로 결정된다.

$$\dot{\lambda}_i(t) = -Q_i x_i(t) - A_i^T \lambda_i(t) - \Pi_i(t) \quad (19)$$

$$\lambda_i(t_f) = 0 \quad (20)$$

식(16)~식(20)에 상위계층의 조정 예측자에서 예측된 $x^o(t), u^o(t), \Pi(t), \beta(t)$ 를 대입하면 선형화된 2점 경계치 문제를 유도할 수 있으며 상태전이 행렬(state transition matrix)의 특성을 이용해 최적의 해를 구해 이를 상위단으로 전달하고 전달된 $x(t), u(t)$ 를 이용해

서 새로운 $\mathbf{x}^o(t)$, $\mathbf{u}^o(t)$, $\Pi(t)$, $\beta(t)$ 의 예측값을 구한다. 이때 다음의 식(21), 식(22)에서 정의된 오차 E_x , E_π 보다 오차가 작아지면 반복계산을 끝내고 최적제어 벡터 $\mathbf{u}(t)$ 를 얻는다. 그러나 오차가 E_x , E_π 보다 크면 만족한 결과를 얻기까지 반복 연산을 수행한다. [6]~[8]

$$\left\{ \int_0^{t_f} (\mathbf{x}^{o(L+1)}(t) - \mathbf{x}^{o(L)}(t))^2 dt \right\}^{\frac{1}{2}} \leq E_x \quad (21)$$

$$\left\{ \int_0^{t_f} (\Pi^{o(L+1)}(t) - \Pi^{o(L)}(t))^2 dt \right\}^{\frac{1}{2}} \leq E_\pi \quad (22)$$

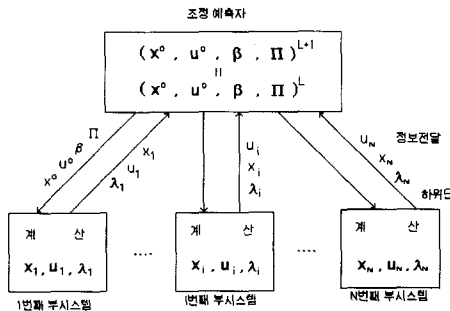


그림 1. 2계층 상호상태 예측
Fig. 1. Two Level Costate Prediction.

그림 1은 제시된 상호예측 알고리즘에 따른 대규모 비선형계의 2계층 상호간의 정보전달 계통도를 보인 것이다.

III. 블록펄스 변환에 의한 대규모 비선형계의 계층별 최적제어

1. 상위계층의 블록펄스 변환

블록펄스 함수의 집합 $Bpf_i(\lambda)$ 은 일반적으로 다음과 같이 단위 구간 $\lambda \in [0,1]$ 에서 정의된다. [9]

$$Bpf_i(\lambda) = \begin{cases} 1, & \frac{i}{m} \leq \lambda < \frac{i+1}{m} \\ 0, & \text{그외구간} \end{cases} \quad (23)$$

단, $i=0,1,2,\dots,m-1$

식(23)의 블록펄스 함수를 이용해서 상위계층 조정 예측자의 $\mathbf{x}^o(t)$, $\mathbf{x}(t)$, $\mathbf{u}^o(t)$, $\mathbf{u}(t)$ 를 각각 전개하면 다음과 같다. [10]

$$\mathbf{x}(t) = \sum_{k=1}^m \mathbf{X}_k \mathbf{B}pf_k(t) \quad (24)$$

$$\mathbf{x}^o(t) = \sum_{k=1}^m \mathbf{X}_k^o \mathbf{B}pf_k(t) \quad (25)$$

$$\mathbf{u}(t) = \sum_{k=1}^m \mathbf{U}_k \mathbf{B}pf_k(t) \quad (26)$$

$$\mathbf{u}^o(t) = \sum_{k=1}^m \mathbf{U}_k^o \mathbf{B}pf_k(t) \quad (27)$$

\mathbf{X}_k , \mathbf{X}_k^o , \mathbf{U}_k , \mathbf{U}_k^o 는 블록펄스 함수 전개시의 계수 벡터이고, 첨자 k 는 블록펄스 함수의 m 항 전개중 k 번째 세부구간을 의미한다.

식(24)에서 식(27)까지를 이용하여 식(7), (8)을 재 표현 하면 블록펄스 함수로 변환된 다음과 같은 식을 얻을 수 있다.

$$[\mathbf{X}_k^o]^{L+1} = [\mathbf{X}_k]^L \quad (28)$$

$$[\mathbf{U}_k^o]^{L+1} = [\mathbf{U}_k]^L \quad (29)$$

여기서 $[\mathbf{X}_k^o]^{L+1}$, $[\mathbf{U}_k^o]^{L+1}$ 의 값은 $L+1$ 번째 반복구간의 비선형 항을 고정시키는데 이용된다. 또한 $\beta(t)$, $\pi(t)$ 의 상위계층의 조정 예측자를 블록펄스 함수로 전개하면 다음과 같다.

$$\beta(t) = \sum_{k=1}^m \mathbf{B}_k \mathbf{B}pf_k(t) \quad (30)$$

$$\pi(t) = \sum_{k=1}^m \mathbf{P}_k \mathbf{B}pf_k(t) \quad (31)$$

그리고 식(10)의 []부분을 블록펄스 함수로 전개 하기 위해 다음과 같이 정의 한다.

$$\left[\frac{\partial \mathbf{f}^T}{\partial \mathbf{x}} \right]_{\mathbf{x}=\mathbf{x}^o, \mathbf{u}=\mathbf{u}^o} \triangleq \begin{bmatrix} f_{11}(t) & f_{12}(t) & \dots & f_{1n}(t) \\ f_{21}(t) & f_{22}(t) & \dots & f_{2n}(t) \\ \vdots & \vdots & \dots & \vdots \\ f_{n1}(t) & f_{n2}(t) & \dots & f_{nm}(t) \end{bmatrix} \triangleq \mathbf{F}(t) \quad (32)$$

식(32)의 각 원소들을 블록펄스 함수로 전개하면 다음과 같다.

$$\begin{aligned} \mathbf{F}(t) &= \sum_{k=1}^m \begin{bmatrix} F_{11k}Bpf_k(t) & F_{12k}Bpf_k(t) & \dots & F_{1nk}Bpf_k(t) \\ F_{21k}Bpf_k(t) & F_{22k}Bpf_k(t) & \dots & F_{2nk}Bpf_k(t) \\ \vdots & \vdots & \dots & \vdots \\ F_{n1k}Bpf_k(t) & F_{n2k}Bpf_k(t) & \dots & F_{nmk}Bpf_k(t) \end{bmatrix} \\ &= \sum_{k=1}^m \mathbf{F}_k \mathbf{B}pf_k(t) \end{aligned} \quad (33)$$

$$F_k = \begin{bmatrix} F_{11k} & F_{12k} & \cdots & F_{1nk} \\ F_{21k} & F_{22k} & \cdots & F_{2nk} \\ \vdots & \vdots & \cdots & \vdots \\ F_{n1k} & F_{n2k} & \cdots & F_{nnk} \end{bmatrix}$$

또한 식(11)의 [] 부분을 같은 방법으로 블록펠스 함수로 전개하면 식(34)와 같다.

$$\left[\frac{\partial f^T}{\partial u} \right]_{x=x^0, u=u^0} \triangleq E(t) = \sum_{k=1}^m E_k B p f_k(t) \quad (34)$$

$$E_k = \begin{bmatrix} E_{11k} & E_{12k} & \cdots & E_{1nk} \\ E_{21k} & E_{22k} & \cdots & E_{2nk} \\ \vdots & \vdots & \cdots & \vdots \\ E_{n1k} & E_{n2k} & \cdots & E_{nnk} \end{bmatrix} \quad (35)$$

이제 $\lambda(t)$ 을 블록펠스 함수로 전개하면 다음과 같다.

$$\lambda(t) = \sum_{k=1}^m \Lambda_k B p f_k(t) \quad (36)$$

따라서 식(10) 및 (11)은 다음과 같은 블록펠스로 변환된 조정 예측자를 얻을 수 있다.

$$[B_k]^{L+1} = [(C_1^T + F_k - A^* T) \Lambda_k]^{L+1} \quad (37)$$

$$[P_k]^{L+1} = [(C_2^T + E_k - B^* T) \Lambda_k]^{L+1} \quad (38)$$

그러므로 식(28), (29), (37), (38)로부터 부시스템의 L+1번째 반복구간의 최적화에 필요한 정보를 알 수 있다.

2. 부시스템의 블록펠스 변환

N개의 독립된 부시스템의 해를 구하기 위해서 변형된 표준형 방정식과 상태전이 행렬에 블록펠스 함수를 적용해 간단한 대수 방정식으로 변환한 후 상위계층에서 얻어진 X_k^0 , U_k^0 , B_k , P_k 를 이용해 필요한 정보인 $x(t)$, $u(t)$, $\lambda(t)$ 를 얻을 수 있다. [12], [13]

먼저 i번째 부시스템의 2점 경계치 문제를 유도하면 다음과 같다. 부시스템 표기를 위한 첨자 i는 생략하고 전개한다.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = L \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} + \begin{bmatrix} M(t) \\ -\pi(t) \end{bmatrix} \quad (39)$$

여기서

$$L = \begin{bmatrix} A & -B R^{-1} B^T \\ -Q & -A^T \end{bmatrix}$$

$M(t) = B(R^{-1}(-\beta(t)) + D(x^0(t), u^0(t), t))$ 이다. 식(39)의 경계조건은 다음과 같다.

$$x(t_0) = x_0 \quad (40)$$

$$\lambda(t_f) = 0$$

상태 천이행렬의 특성을 이용해서 식(39)의 해를 유도하면 다음과 같다.

$$\begin{bmatrix} x(t_f) \\ \lambda(t_f) \end{bmatrix} = \Phi(t_f, t) \begin{bmatrix} x(t) \\ \lambda(t) \end{bmatrix} + \int_t^{t_f} \Phi(t_f, \tau) \begin{bmatrix} M(\tau) \\ -\pi(\tau) \end{bmatrix} d\tau \quad (41)$$

위 식의 적분부분을 아래와 같이 정의한다.

$$\begin{bmatrix} G_1(t) \\ G_2(t) \end{bmatrix} \triangleq \int_t^{t_f} \Phi(t_f, \tau) \begin{bmatrix} M(\tau) \\ -\pi(\tau) \end{bmatrix} d\tau \quad (42)$$

위 식을 이용하여 식(41)의 $\lambda(t_f)$ 부분만을 다시 전개하면 다음과 같다.

$$\lambda(t_f) = \Phi_{21}(t_f, t) x(t) + \Phi_{22}(t_f, t) \lambda(t) + G_2(t) \quad (43)$$

식(40)의 경계조건을 이용하여 $\lambda(t)$ 을 구하면 다음과 같다.

$$\lambda(t) = -\Phi_{22}^{-1}(t_f, t) \Phi_{21}(t_f, t) x(t) - \Phi_{22}^{-1}(t_f, t) G_2(t) \quad (44)$$

이제 식(39)는 다음과 같이 된다.

$$\dot{x}(t) = A x(t) - B R^{-1} B^T \lambda(t) + M(t) \quad (45)$$

식(44)를 식(45)에 대입하여 정리하면 다음과 같다.

$$\begin{aligned} \dot{x}(t) &= A x(t) + B R^{-1} B^T \Phi_{22}^{-1}(t_f, t) \Phi_{21}(t_f, t) x(t) \\ &\quad + M(t) + B R^{-1} B^T \Phi_{22}^{-1}(t_f, t) G_2(t) \\ &= F(t) x(t) + k(t) \end{aligned} \quad (46)$$

$$\begin{aligned}
 F(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \Phi_{22}^{-1}(t_f, t) \Phi_{21}(t_f, t) \\
 \mathbf{k}(t) &= \mathbf{M}(t) + \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \Phi_{22}^{-1}(t_f, t) \mathbf{G}(t)_2
 \end{aligned}
 \tag{47}$$

상태전이 행렬을 이용한 처리 과정에 블록펄스 함수를 적용하기 위해 특성식에 블록펄스 함수를 적용하면 다음과 같다.

$$\begin{aligned}
 \Phi_m &= \left[\mathbf{I} - \left(\frac{t_f}{2m} \right) \mathbf{L} \right]^{-1} \\
 \Phi_k &= \Phi_{k+1} \left[\mathbf{I} + \left(\frac{t_f}{2m} \right) \mathbf{L} \right] \left[\mathbf{I} - \left(\frac{t_f}{2m} \right) \mathbf{L} \right]^{-1}
 \end{aligned}
 \tag{48}$$

여기서 $k=m-1, m-2, \dots, 1$ 이다.

또한, $\mathbf{M}(t), \mathbf{G}_2(t), \mathbf{F}(t), \mathbf{k}(t)$ 를 블록펄스 함수로 전개하면 다음과 같다.

$$\mathbf{M}(t) = \sum_{k=1}^m \mathbf{h}_k \mathbf{Bpf}_k(t) \tag{49}$$

$$\mathbf{G}_2(t) = \sum_{k=1}^m \mathbf{g}_{2k} \mathbf{Bpf}_k(t) \tag{50}$$

$$\mathbf{F}(t) = \sum_{k=1}^m \nu_k \mathbf{Bpf}_k(t) \tag{51}$$

$$\mathbf{k}(t) = \sum_{k=1}^m \mathbf{w}_k \mathbf{Bpf}_k(t) \tag{52}$$

식(42)에 블록펄스 함수의 역방향 적분을 취하면 다음과 같다.

$$\begin{aligned}
 \mathbf{g}_{2m} &= \left(\frac{t_f}{2m} \right) [\Phi_{21m} \mathbf{h}_m + \Phi_{22m} \mathbf{z}_m] \\
 \mathbf{g}_{2k} &= \mathbf{g}_{2k+1} + \left(\frac{t_f}{2m} \right) [\Phi_{21k} \mathbf{h}_k + \Phi_{22k} \mathbf{z}_k] \\
 &\quad + \left(\frac{t_f}{2m} \right) [\Phi_{21k+1} \mathbf{h}_{k+1} + \Phi_{22k+1} \mathbf{z}_{k+1}]
 \end{aligned}
 \tag{53}$$

여기서 $k=m-1, m-2, \dots, 1$ 이다. 식(47)에 블록펄스 함수를 적용하면 각각의 계수벡터는 다음과 같다.

$$\begin{aligned}
 \nu_k &= \mathbf{A} + \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \Phi_{22k}^{-1} \Phi_{21k} \\
 \mathbf{w}_k &= \mathbf{h}_k + \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \Phi_{22k}^{-1} \mathbf{g}_{2k}
 \end{aligned}
 \tag{54}$$

여기서 $k=1, 2, 3, \dots, m$ 이다. 결국, 식(46)의 해는 다음의 대수 반복연산만으로 얻어진다.

$$\begin{aligned}
 \mathbf{x}_1 &= \left[\mathbf{I} - \left(\frac{t_f}{2m} \right) \nu_1 \right]^{-1} \left[\mathbf{x}_0 + \left(\frac{t_f}{2m} \right) \mathbf{w}_1 \right] \\
 \mathbf{x}_{k+1} &= \left[\mathbf{I} - \left(\frac{t_f}{2m} \right) \nu_k \right]^{-1} \\
 &\quad \left[\left(\mathbf{I} + \left(\frac{t_f}{2m} \right) \nu_k \right) \mathbf{x}_k + \left(\frac{t_f}{2m} \right) (\mathbf{w}_k + \mathbf{w}_{k+1}) \right]
 \end{aligned}
 \tag{55}$$

여기서 $k=1, 2, \dots, m$ 이다.

식(55)로부터 $\mathbf{x}(t)$ 의 해를 구하면 식(44)로부터 $\lambda(t)$ 를 구할 수 있으며 잠정적인 최적제어 벡터 $\mathbf{u}(t)$ 을 구할 수 있게 된다. 그런데, 각 부시스템이 잠정적인 최적제어 입력이 전체 시스템의 최적제어 입력인가를 확인하기 위해서 오차 함수를 다음과 같이 정의한다.

$$\begin{aligned}
 \mathbf{E}_x &= \left[\sum_{i=1}^N \sum_{j=1}^m \{ (\mathbf{x}_i)_j^{L+1} - (\mathbf{x}_i)_j^L \}^T \{ (\mathbf{x}_i)_j^{L+1} - (\mathbf{x}_i)_j^L \} \right]^{\frac{1}{2}} \\
 \mathbf{E}_\pi &= \left[\sum_{i=1}^N \sum_{j=1}^m \{ (\pi_i)_j^{L+1} - (\pi_i)_j^L \}^T \{ (\pi_i)_j^{L+1} - (\pi_i)_j^L \} \right]^{\frac{1}{2}} \\
 \mathbf{E} &= \frac{\mathbf{E}_x + \mathbf{E}_\pi}{m}
 \end{aligned}
 \tag{56}$$

여기서 i 는 i 번째 부시스템, j 는 j 번째 구간, $L+1$ 은 L 의 다음 단계이다. 각각의 계층에서 블록펄스 함수 계수벡터들의 대수 연산만을 이용하므로 알고리즘이 간단하게 정리됨을 알 수 있다.

IV. 시뮬레이션

무한모선(infinite bus-bar)에 연결된 비선형 동기 기 시스템으로 대부분의 시간을 정상상태로 동작을 하지만 수용가 측의 부하변동이나 단락사고가 발생했을 때 동기기의 계자전압을 이용해 최적인 방법으로 시스템을 원상태로 되돌리는 문제를 고려한다. 이 비선형 시스템은 Mukhopadhaya 및 Malik에 의해 식(57)과 같이 모델링 되었다.^[14]

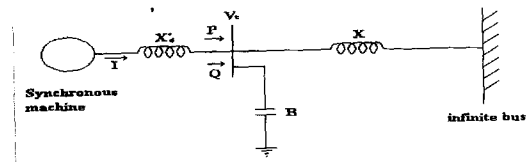


그림 2. 무한 모선에 연결된 동기기
Fig. 2. Synchronous machine connected to infinite bus.

$$\begin{cases}
 \dot{y}_1(t) = y_2(t) \\
 \dot{y}_2(t) = b_1 - a_1 y_2(t) - a_2 y_3(t) \sin(y_1(t)) \\
 \quad - \frac{b_2}{2} \sin(2y_1(t)) \\
 \dot{y}_3(t) = u_1(t) - c_1 y_3(t) + c_2 \cos(y_1(t))
 \end{cases}
 \tag{57}$$

시스템의 각 상태변수, 제어변수의 물리적 의미는 다음과 같다.

- $y_1(t)$: 회전자변위(rotor angle) δ , [radians]
- $y_2(t)$: 회전자속도(speed deviation) $\dot{\delta}$, [rad / sec]
- $y_3(t)$: 계자쇄교자속(field flux linkage) ψ_f
- $u_1(t)$: 동기기 계자전압(applied voltage) v_f

또한 파라미터 값은 식(58)과 같다. 여기서 S와 H는 고려되지 않음을 알 수 있다.

$$a_1 = 0.2703, \quad a_2 = 12.0120, \quad b_1 = 39.1892, \\ b_2 = -48.0480 \quad c_1 = 0.3222, \quad c_2 = 1.9000$$

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix} \quad (58)$$

시스템의 동작점은 식(59)로, 초기치는 식(60)과 같다.

$$y_{1e}=0.7641 \text{ [radians]}, \quad y_{2e}=0 \text{ [rad/s]}, \\ y_{3e}=7.7438, \quad u_e=1.1 \quad (59)$$

$$y_0 = [0.6461 \quad 0 \quad 7.7438]^T \quad (60)$$

이제 평가함수는 식(61)로 표현된다.

$$J = \frac{1}{2} \int_0^{t_f} [q_1 (y_1 - y_{1e})^2 + q_2 (y_2 - y_{2e})^2 \\ + q_3 (y_3 - y_{3e})^2 + r (u - u_e)^2] dt \quad (61)$$

여기에 식(58)~식(60)의 파라미터 값을 적용하면 J는 식(62)와 같이 된다.

$$J = \frac{1}{2} \int_0^{t_f} \{ (y_1 - 0.7461)^2 + y_2^2 + 0.1(y_3 - 7.7438)^2 \\ + 100(u_1 - 1.1)^2 \} dt \quad (62)$$

식(57)의 비대각 선형화 행렬은 다음과 같다.

$$A^* = \left[\frac{\partial f}{\partial y} \right]_{y=y_e} = \begin{bmatrix} 0 & 1 & 0 \\ -64.5348 & -0.2703 & -8.1533 \\ -1.2896 & 0 & -0.3222 \end{bmatrix} \\ B^* = \left[\frac{\partial f}{\partial u} \right]_{u=u_1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (63)$$

$y - y_e = x$, $u_1 - u_e = u$ 로 두면 식(57)을 다음과 같이 쓸 수 있다.

$$\dot{x}(t) = A^* x(t) + B^* u(t) + f(y, u_e, t) - A^* x(t) \quad (64)$$

선형화된 행렬 A^* , B^* 를 대각행렬과 비대각행렬로 다시 분할하면 다음과 같다.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -64.5348 & -0.2703 & 0 \\ 0 & 0 & -0.3222 \end{bmatrix} \\ B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -8.1533 \\ -1.2896 & 0 & 0 \end{bmatrix} \quad (65)$$

N개의 시스템 분할을 위해 상위계층의 $x = x^o$ 을 이용하여 비대각, 비선형항을 $D(x^o, t)$ 로 정의하면 식(64)는 다음과 같이 간략히 된다.

$$\dot{x}(t) = A x(t) + B u(t) + D(x^o, t) \quad (66)$$

이제 시스템을 2개의 부시스템으로 다음과 같이 분할하여 시뮬레이션을 수행한다.

(1) 1번째 부시스템

$$\dot{x}_{s_1}(t) = A_{s_1} x_{s_1}(t) + B_{s_1} u_{s_1}(t) + D_{s_1}(x^o, t)$$

$$J_1 = \frac{1}{2} \int_0^{t_f} (\| x_{s_1}(t) \|_{Q_{s_1}}^2 + \| u_{s_1}(t) \|_{R_{s_1}}^2) dt$$

여기서 $Q_{s_1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R_{s_1} = 100$ 이다.

(2) 2번째 부시스템

$$\dot{x}_{s_2}(t) = A_{s_2} x_{s_2}(t) + B_{s_2} u_{s_2}(t) + D_{s_2}(x^o, t)$$

$$J_2 = \frac{1}{2} \int_0^{t_f} (\| x_{s_2}(t) \|_{Q_{s_2}}^2 + \| u_{s_2}(t) \|_{R_{s_2}}^2) dt$$

여기서 $Q_{s_2} = [1]$, $R_{s_2} = 100$ 이다.

다음은 예제에 대한 시뮬레이션 결과들로 오차는 $E = 1.0 \times 10^{-3}$ 로 설정하여 수행했다. 먼저 그림 3~6은 최적화 구간을 $t_f = 2$ [sec]로 전개항수를 $m = 50$ 으로 두고 준선형화 기법과 본 연구 방법에 의한 결과와의 객관적 비교를 보인 것으로 본 연구 방법의 결과가 주어진 값에 잘 추종함을 알 수 있다.

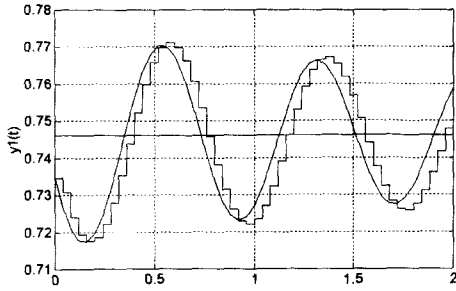


그림 3. 최적 상태벡터 $y_1(t)$ 의 비교
Fig. 3. Comparison for optimal state $y_1(t)$.

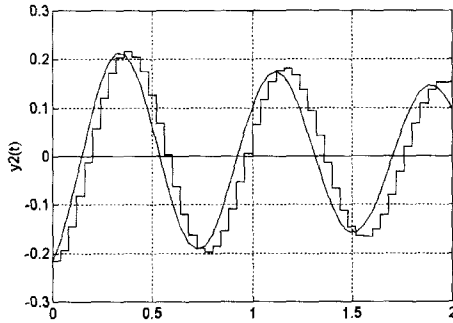


그림 4. 최적 상태벡터 $y_2(t)$ 의 비교
Fig. 4. Comparison for optimal state $y_2(t)$.

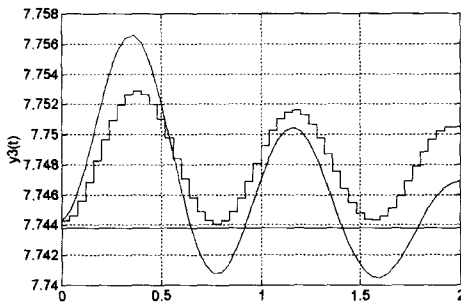


그림 5. 최적 상태벡터 $y_3(t)$ 의 비교
Fig. 5. Comparison for optimal state $y_3(t)$.

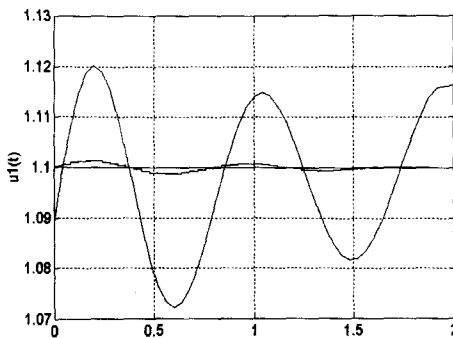


그림 6. 최적 제어벡터 $u(t)$ 의 비교
Fig. 6. Comparison for optimal control $u(t)$.

아래의 표 1은 기존의 방법인 평가함수에 구속조건을 고려한 경우와 개선된 본 연구 방법과의 각 경우에 따른 결과를 비교한 것으로 기존의 방법에서 <경우1>은 행렬 $S=0.2$, $R=100$ 으로, <경우2>는 $S=0.2$, $R=50$ 으로 설정한 것이며 본 연구 방법은 <경우1, 2> 모두 $S=0$ 이면서 $R=100, 50$ 으로 한 경우이다. 여기서 L 은 반복지표, E 는 오차를 의미한다. 표에서 알 수 있듯이 기존의 방법은 수렴조건을 고려치 않으므로 시행착오법 등을 통해 행렬 S 를 결정하나 최적의 값을 결정하기가 쉽지 않으며 실제로 S 가 적절히 설정되지 않으면 <경우1>에서처럼 거의 무한 반복에 가까운 반복지표를 가져 결과적으로 실시간 처리가 불가능함을 알 수 있으며 <경우2>에서는 R 값의 변화와 함께 오차가 무한대에 가까워져 아예 수렴하지를 못함을 알 수 있다. 반면 본 연구 방법에서 제시한 것처럼 수렴조건을 고려해서 평가함수에 구속조건을 부여하지 않는 경우 R 값에 관계없이 <경우1, 2> 모두 반복지표나 수렴정도에 있어서 만족할 만한 결과를 보이고 있다. 결과적으로 기존의 방법이 갖는 알고리즘 상의 단점이 입증되었으며 본 연구 방법에 의해 제시된 알고리즘이 기존의 방법이 갖는 입증된 단점을 근본적으로 제거함과 동시에 결과도 매우 양호함을 알 수 있다.

표 1. 파라미터 설정에 따른 수렴정도 비교
Table 1. Comparison for convergence according to parameter.

| 알고리즘 파라미터 | 기존의 방법 | | 본 연구 방법 | |
|--------------------------------|---------------------------|-----------------------------------|--------------------------|--------------------------|
| | 경우1 | 경우2 | 경우1 | 경우2 |
| $t_f=50(\text{sec})$ $m=50$ | $L=577$ $E=9.9948e-04$ | $L=80$ $E=\text{inf.}(\infty)$ | $L=39$ $E=8.1417e-04$ | $L=65$ $E=8.9645e-04$ |

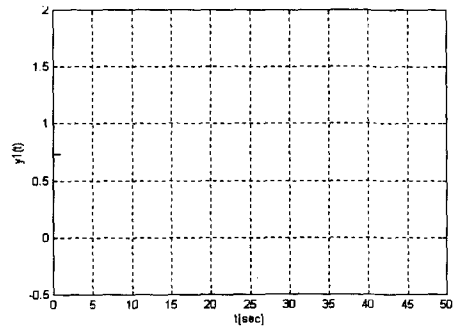


그림 7. 기존 방법의 최적 상태벡터 $y_1(t)$
Fig. 7. optimal state $y_1(t)$ by previous method.

그림 7과 그림 8은 위의 각 방법의 <경우2>에 대한 최적 상태벡터 $y_1(t)$ 에 대한 것으로 기존의 방법에

의한 그림 7에서는 최적화 구간 초기에 0.7347의 값을 가진 후 수렴하지 못하는 관계로 그림 상에 궤적이 나타나지 못하나 그림 8은 본 연구 방법에 의한 것으로 잘 수렴하고 있음을 알 수 있다.

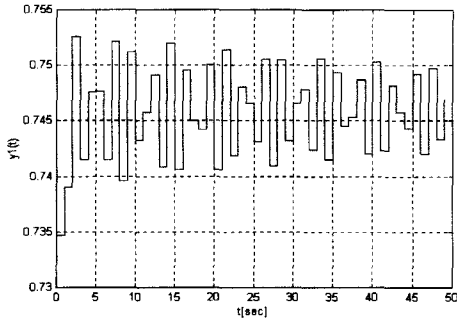


그림 8. 본 연구 방법의 최적 상태벡터 $y_1(t)$
Fig. 8. optimal state $y_1(t)$ by new method.

표 2는 기존의 방법과 본 연구 방법간의 최적화 구간에 따른 반복지표 및 오차의 수렴정도를 비교한 것으로 기존의 방법에서 제시한 대로 행렬 $S=0.1$ 로 두고 최적화 구간을 $t_f=2$ [sec]에서 50 [sec]으로 확장해간 결과 기존의 방법은 반복지표 L 이 급격히 증가하며 수렴속도가 현저히 떨어짐을 알 수 있는데 이는 기존의 방법이 바람직한 수렴 결과를 제한된 최적화 구간에서만 얻을 수 있으며 실시간 처리에도 적절치 못함을 나타내고 있다. 반면 본 연구 방법은 전 최적화 구간에서 기존의 방법에 비해 결과가 양호함은 물론 최적화 구간의 확장에도 불구하고 알고리즘의 반복지표가 전체적으로 적음을 알 수 있다. 특히 $t_f=50$ [sec]에서 반복지표가 기존의 방법에 비해 43.8%로 상대적으로 현저히 적은 양호한 결과를 보이며 수렴속도 면에 있어서도 2.3배의 빠른 상대속도를 보임으로써 본 연구 방법에 의한 개선된 알고리즘이 기존의 방법이 갖는 제한된 최적화 구간에서의 문제를 해결함과 동시에 큰 폭의 수렴속도 향상을 가져왔음을 알 수 있다. 또한 그림 9는 본 연구 방법을 이용해서 최적화 구간 $t_f=50$ [sec], 전개항수 $m=200$ 으로 증가한 경우의 결과를, 그림 10은 $t_f=50$ [sec]일 때 $m=50$ 및 100의 결과를 비교한 것이다. 그림 9, 10에서 모두 최적화 구간이 증가해도 주어진 값에 잘 수렴됨을 확인할 수 있으며 그림 10은 전개항수를 증가함에 따라 오차가 감소됨을 보여주고 있는데 이는 그림 3~6에서 나타난 시스템 분할에 따른 위상 차를 최소화하는데

기여할 수 있을 것이다.

표 2. 최적화 구간에 따른 반복지표와 오차 값 비교

Table 2. Comparison for iteration index and error.

| 전개항수 (m) | 최적화 구간 t_f (sec) | 기존의 방법 | | 본 연구 방법 | |
|----------|--------------------|---------|----------------|---------|----------------|
| | | 반복지표(L) | 오차 | 반복지표(L) | 오차 |
| 50 | 2 | 6 | $E=4.5529e-04$ | 7 | $E=2.1302e-04$ |
| | 12 | 12 | $E=4.0766e-04$ | 11 | $E=5.9004e-04$ |
| | 20 | 14 | $E=3.8285e-04$ | 13 | $E=6.4347e-04$ |
| | 50 | 89 | $E=9.9696e-04$ | 39 | $E=8.1417e-04$ |

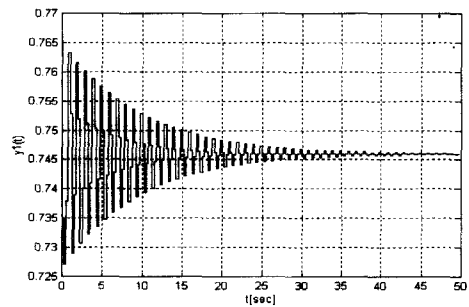


그림 9. 최적 상태벡터 $y_1(t)$
Fig. 9. Optimal state $y_1(t)$.

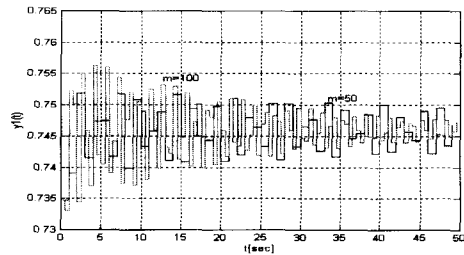


그림 10. 최적 상태벡터 $y_1(t)$ 의 비교
Fig. 10. Comparison for optimal state $y_1(t)$.

V. 결론

본 연구에서는 대규모 비선형 시스템을 최적화하기 위한 새로운 개선된 알고리즘을 제시하고 비선형 시스템인 무한 모션에 연결된 동기에 제시된 알고리즘을 적용하여 만족할 만한 결과를 얻었다. 이를 위해 Singh와 Hassan이 제시한 비선형 시스템을 분할하는 계층별 최적제어 방법을 이용했다. 그러나 기존의 알고리즘은 1)행렬 Q, R, S, H 를 결정하기 위한 별도

의 과정이 요구되며 설정된 행렬 값에 따라서는 수렴하지 못하는 결과를 나타내며, 2)평가함수에 구속 조건을 고려해야 하며, 3)제한된 최적화 구간에서만 바람직한 결과를 얻을 수 있으며, 4)반복연산 수행 회수의 현저한 증가로 수렴속도가 현저히 떨어져 실시간 처리에 어려움이 있는 등 알고리즘 상의 문제가 있다. 따라서 본 연구에서는 알고리즘 설계에 수렴조건 k 를 고려해서 1)시스템 행렬 결정을 위한 과정이 없으며 2)평가함수에 구속조건을 부여하지 않고 최적의 수렴성 및 최적화 구간을 얻도록 새로운 개선된 알고리즘을 제시함으로써 기존의 알고리즘이 갖는 문제점을 근본적으로 해결하였으며 최적화 구간 t_1 의 확장과 관련하여 3)본 연구 방법은 상대적으로 현저히 적은 반복 지표를 갖는 양호한 결과를 보여 실시간 처리에서 향상된 결과를 보였다. 4)또한 최적화 구간이 증가해도 최적 상태케적이 주어진 명령 값에 잘 추종함을 알 수 있어 수렴속도와 함께 수렴성이 우수함이 입증되었다. 따라서 본 연구 방법에 의한 개선된 대규모 비선형 시스템의 계층별 최적제어 기법이 기존의 방법에 비해 알고리즘 상의 우수함을 알 수 있었다. 또한 블록펄스 함수 변환을 통해 계산이 용이하며 컴퓨터 처리 용량을 적게 차지하는 장점이 있음을 알 수 있었다.

참 고 문 헌

- [1] Janishi, "Optimal Control of Non-Linear Power Systems by an Imbedding Method", *Automatica*, Vol. 11, pp. 633-636, 1975.
- [2] A.P. Sage, "Optimal System Control", Prentice-Hall, Inc., New Jersey, pp. 192-208, 1977.
- [3] M. G. Singh and M. Hassan, "The Optimization of Non-Linear Systems using a New Two Level Method". *Automatica*, Vol. 12, pp. 359-363, 1976.
- [4] M.G.Singh, "Dynamical Hierarchical Control", *Automatica*, Vol. 18, No.4, pp. 501-503, 1982.
- [5] M. G. Singh, "Dynamical Hierarchical Control", North-Holland Publishing Co., pp.61-73, 1980.
- [6] M. Hassan, M. S. Mahmoud, M. G. Singh and M. P. Spathopolous, "A Two-level Parameter Estimation Algorithm Using the Multiple Projection Approach". *Automatica*. Vol. 18, No. 5, pp. 621-630, 1982.
- [7] M. F. Hassan and M. G. Singh, "Hierarchical Successive Approximation Algorithms for Non-linear Systems. Part I. Generalisation of the Method of Takahara", *Large Scale Systems*, Vol. 2, pp. 65-79, 1981.
- [8] M. F. Hassan and M. G. Singh, "Hierarchical Successive Approximation Algorithms for Non-linear Systems. Part II. Algorithm Based on Costate Coordination", *Large Scale Systems*, Vol. 2, pp. 81-95, 1981.
- [9] K.R. Palanisamy and D. K. Bhattacharya, "System identification via block-pulse functions". *Int. J. Systems Sci.*, Vol. 12, No. 5, pp. 643-647, 1981.
- [10] Chyi Hwang and Yen-Ping Shih. "On The Operational Matrices of Block Pulse Functions". *Int. J. Systems Sci.*, Vol. 17, No. 10, pp. 1489-1498, 1986.
- [11] 이한석, 안두수, "블록펄스함수에 의한 비선형 계의 2계층 최적제어", 대한전기학회 논문집 제47권 4호 pp. 494-502, 1998
- [12] Jian-Min Zhu and Yong-Zai Lu. "Hierarchical Optimal Control For Distributed Parameter Systems Via Block Pulse Operator". *Int. J. Control*, Vol. 48, No. 2, pp. 685-703, 1988.
- [13] Jian-Min Zhu and Yong-Zai Lu. "Hierarchical Strategy For Non-Linear Optimal Control Systems via STWS Approach". *Int. J. Control*, Vol. 47, No. 6, pp. 1837-1848, 1988.
- [14] M.G.Singh M. Hassan and J. L. Calvet. "Hierarchical Optimization of Non-Linear Systems With Application To A Synchronous Machine". *Int. J. Systems Sci.*, Vol. 7, No. 9, pp. 1041-1051, 1976.

저 자 소 개



朴 俊 勳(正會員)

1957년 3월 25일생. 1991년 삼성
전자 기술교육과장. 1992년 성균
관대학교 대학원 전기공학과 졸
업(공학). 1997년 ~ 1998년 영국
레스터대 방문교수 현재 충주대
학교 제어계측공학과 조교수

金 鐘 夫(正會員) 第35卷 T編 2號 參照