

# 大型動的시스템의 最適化 알고리즘 및 프로그램 開發에 관한 研究

|        |
|--------|
| 論 文    |
| 32~4~2 |

## Algorithms and Programs for Optimization of Large-Scale Dynamic System

梁 興 錫\* · 朴 永 文\* · 金 建 中\*\*  
(Heung-Suk Yang · Young-Moon Park · Kun-Joog Kim)

### Abstract

In this paper an efficient algorithm for Pontriagin's maximum principle is developed. Fletcher-Powell method is adopted as optimization technique which shows fast and stable convergence characteristics. Terminal constraints are also considered by using Hestens' algorithm and penalty function method together. Control variable inequality constraints are also considered by using Gradient Projection technique combined with Fletcher-Powell method. Test experiment shows good and reliable results.

### 1. 緒 論

動的系統의 最適制御 問題는 폰트리아긴의 最大原理 (maximum principle of Pontriagin)에 의해 最適條件이 表示된다. 그러나 最大原望는 解析的으로는 動的系統의 最適制御 問題를 간결히 表示할 수 있으나, 실제로 數值計算을 할 때에는 많은 어려움이 따른다.

이것은 系統程式이 일반적인 最適化問題에서의 等號制約條件에 해당되기 때문이다. 즉 전 区間에 걸쳐서 系統方程式으로 주어지는 等號制約條件이 存在할 뿐만 아니라, 이것이 매 区間마다 서로 연계가 되어 있기 때문에 일반적인 等號制約條件과는 그 特性이 다르며 더욱 복잡한 경우가 된다고 생각하면 된다.

따라서 이에관한 解法은 일반적으로 간단히 처리되지 않고 또한 計算時間이 많이 걸리게 된다.

특히 終端點이 고정된 경우에는 問題는 더욱 복잡하여진다. 이것도 또한 等號制約條件으로 간주를 할 수가 있으나 이 경우에는 全区間의 結果가 終端點에 영향을 미치게 되므로 每区間을 獨立的으로 解析할 수 없게 된다. 따라서 이때에는 逐次的으로 수렴을 시켜나가야만 發散하는 問題를 피할 수 있게 된다. 일반적인 終端點 처리 方法에는 스위프法(Sweep method), 페널티函數法, 헤스테네스法 (Hestens' method) 등이 있다.本 研究에서는 페널티函數法과 헤스테네스法을 병용하여 使用하였다. 이것은 스위프<sup>1), 2), 6)</sup>에서와 같은 많은 기억용량이 必要하지 않고 간단히 反復的으로 처리할 수 있는 반면 또한 헤스테네스法<sup>3)</sup>의 短點인 振動과 發散의 問題를 同時에 해결할 수 있었다.

制御變數에 不等號制約條件이 있는 問題는 라그랑주乘數를 도입하여 K-T條件 (Kuhn-Tucker condition)에 의해 해결하거나 혹은 傾斜投影法 (Gradient projection method)에 의해 처리할 수 있다. 여기서는 傾斜投影法을 使用하였고 또한 플레처-포웰<sup>5)</sup> (Fletcher-Powell)法

\* 正 會 員 : 서울大 工大 電氣工學科 教授 · 工博  
\*\* 正 會 員 : 忠南大 工大 電氣工學科 助教授  
接受日字 : 1982年10月27日

을 도입하여 전체적인 수렴속도를 크게 改善하였다.

## 2. 本 論

### 1. 이산형 계통의 最大原理

일반적인 動的系統의 最適制御問題는 연속계통으로 表示가 된다. 그러나 現代에는 計算機의 發展에 힘입어 이산계통으로 처리를 할 수있게 되었다. 연속계통의 問題는 간단히 이산계통으로 모델링이 가능하므로 本研究에서는 이산계통의 最適制御問題만을 다루기로 한다.

이산계통의 最適制御問題는 다음과 같이 表示할 수 있다.

$$\begin{aligned} \mathbf{X}(i+1) &= f^i [ \mathbf{X}(i), \mathbf{u}(i) ]; \\ \mathbf{X}(0) & \text{ given}, i = 0, \dots, N-1 \quad (1) \\ \min J &= \phi [ \mathbf{X}(N) ] + \sum_{i=0}^{N-1} L^i [ \mathbf{X}(i), \mathbf{u}(i) ] \quad (2) \end{aligned}$$

단  $\mathbf{X}$ 는  $n$ 次元 벡터  
 $\mathbf{u}$ 는  $m$ 次元 制御벡터

$f^i$ 는 系統方程式  
 $L^i$ 는  $i$ 區間의 평가函數  
 問題는 式(1)로써 表示되는 系統에서 式(2)를 最小로 하는 制御벡터  $\mathbf{u}$ 를 決定하는 것이다. 式(1)은 等號制約條件으로 볼 수 있으므로 라그랑류 乘數  $\lambda$ 를 도입하여 式(2)에 확장시키면

$$\begin{aligned} \bar{J} &= \phi [ \mathbf{X}(N) ] + \sum_{i=0}^{N-1} \{ L^i [ \mathbf{X}(i), \mathbf{u}(i) ] + \lambda^T(i+1) \{ f^i [ \mathbf{X}(i), \mathbf{u}(i) ] - \mathbf{X}(i+1) \} \} \quad (3) \end{aligned}$$

편의상 스칼라函數  $H^i$ 를 도입하자.  
 $H^i = L^i [ \mathbf{X}(i), \mathbf{u}(i) ] + \lambda^T(i+1) f^i [ \mathbf{X}(i), \mathbf{u}(i) ], i = 0, \dots, N-1 \quad (4)$

式(3)을  $H^i$ 를 使用하여 다시 정리하면

$$\begin{aligned} \bar{J} &= \phi [ \mathbf{X}(N) ] - \lambda^T(N) \mathbf{X}(N) + \sum_{i=1}^{N-1} \{ H^i - \lambda^T(i) \mathbf{X}(i) \} + H^0 \quad (5) \end{aligned}$$

式(5)에서  $\mathbf{u}(i)$ 에 대한  $\bar{J}$ 의 변분을 보면

$$d\bar{J} = [ \frac{\partial \phi}{\partial \mathbf{X}(N)} - \lambda^T(N) ] d\mathbf{X}(N) + \sum_{i=1}^{N-1} [ \frac{\partial H^i}{\partial \mathbf{X}(i)} d\mathbf{X}(i) + \lambda^T(i) d\mathbf{X}(i) + \frac{\partial H^i}{\partial \mathbf{u}(i)} d\mathbf{u}(i) ] + \frac{\partial H^0}{\partial \mathbf{u}(0)} d\mathbf{u}(0) \quad (6)$$

이때  $\lambda$ 를 다음과 같이 決定하자.  
 $\lambda^T(i) = \frac{\partial H^i}{\partial \mathbf{X}(i)} = \frac{\partial L^i}{\partial \mathbf{X}(i)} + \lambda^T(i+1) \frac{\partial f^i}{\partial \mathbf{X}(i)}, i = 0, \dots, N-1 \quad (7)$

$$\lambda^T(N) = \frac{\partial \phi}{\partial \mathbf{X}(N)} \quad (8)$$

따라서 式(6)은 다음과 같이 表示된다.  
 $d\bar{J} = \sum_{i=1}^{N-1} \frac{\partial H^i}{\partial \mathbf{u}(i)} d\mathbf{u}(i) + \lambda^T(0) d\mathbf{X}(0) \quad (9)$

$\mathbf{X}(0)$ 는 주어졌으므로  $d\mathbf{X}(0) = 0$ 이다. 따라서  $\frac{\partial H^i}{\partial \mathbf{u}(i)}$ 는 式(7), (8)의 條件에서  $\bar{J}$ 의  $\mathbf{u}(i)$ 에 대한 傾斜가 된다. 그러므로 最適條件은

$$\frac{\partial H^i}{\partial \mathbf{u}(i)} = 0, i = 0, \dots, N-1 \quad (10)$$

윗식을 다시 정리하면  
 $\mathbf{X}(i+1) = f^i [ \mathbf{X}(i), \mathbf{u}(i) ] \quad (11)$

$$\lambda(i) = [ \frac{\partial f^i}{\partial \mathbf{X}(i)} ]^T \lambda(i+1) + [ \frac{\partial L^i}{\partial \mathbf{X}(i)} ]^T \quad (12)$$

$$\frac{\partial H^i}{\partial \mathbf{u}(i)} = \frac{\partial L^i}{\partial \mathbf{u}(i)} + \lambda^T(i+1) \frac{\partial f^i}{\partial \mathbf{u}(i)} = 0 \quad (13)$$

윗식에서 境界條件은 다음과 같이 분리되어 있다.  
 $\mathbf{X}(0)$  given (14)  
 $\lambda(N) = [ \frac{\partial \phi}{\partial \mathbf{X}(N)} ]^T \quad (15)$

이와같이 境界條件이  $i = 0$ 과  $i = N$ 에 분리되어 있는 問題를 2點境界值問題라 하며 풀기가 어려운 點이 있다.

本研究에서는 이와같은 2點境界值問題를 境界條件을 自動적으로 만족하면서 평가함수를 最小로 하는 制御벡터  $\mathbf{u}(i)$ 를 決定하고자 한다. 즉 變數로서는  $\mathbf{u}(i)$ 만을 取하고 式(14), (15)의 條件下에서 式(11), (12)를 풀어서  $\mathbf{X}(i)$ 와  $\lambda(i)$ 를 決定하고 最適條件式(13)을 만족하도록 한다. 이때 最適化技法으로는 플레처 - 포웰法을 使用하였다. 이것은 평가函數가 아무리 간단한 경우일지라도

실제는 式(11), (12)에 의해 内容上으로는  $\mathbf{u}(i)$ 에 對해 매우 복잡한 형태가 되기 때문이다. 實驗 決果에 의하면 傾斜法을 使用하면 그 수렴 속도가 매우 느린것을 볼 수 있었다.

플레처-포웰법<sup>7)</sup>의 알고리즘은 다음과 같다.

i)  $\mathbf{u}^0$ 을 가정

ii)  $\min_{\mathbf{u}} \bar{J}(\mathbf{X}, \mathbf{u}, \boldsymbol{\lambda})$  (플레처-포웰법)

단,  $\mathbf{X}(i+1) = f^i(\mathbf{X}(i), \mathbf{u}(i))$ ,  $\mathbf{X}(0)$  given

$$\boldsymbol{\lambda}(i) = \left[ \frac{\partial f^i}{\partial \mathbf{X}(i)} \right]^T \boldsymbol{\lambda}(i+1) + \left[ \frac{\partial L^i}{\partial \mathbf{X}(i)} \right]^T,$$

$$\boldsymbol{\lambda}(N) = \left[ \frac{\partial \phi}{\partial \mathbf{X}(N)} \right]^T$$

iii)  $\| \frac{\partial H^i}{\partial \mathbf{u}(i)} \| \leq \epsilon$  이면 計算끝  
 $\geq \epsilon$  Go To ii)

## 2. 終端條件의 처리

終端에 상태변수의 函数로 된 條件이 存在할 때에는 問題가 더욱 복잡하여진다. 이때의 問題는 다음과 같이 定式化된다.

$$\min_{\mathbf{u}} J = \phi(\mathbf{X}(N)) + \sum_{i=1}^{N-1} L^i(\mathbf{X}(i), \mathbf{u}(i)) \quad (16)$$

$$\mathbf{X}(i+1) = f^i(\mathbf{X}(i), \mathbf{u}(i)) \quad (17)$$

$$\boldsymbol{\psi}(\mathbf{X}(N)) = 0 \quad (18)$$

여기서  $\boldsymbol{\psi}$ 는  $q$  次的 벡터函数이고  $q \leq N$ 인 관계가 있다. 이때  $\boldsymbol{\psi}$ 도 또한 等號制約條件으로 볼 수 있다. 따라서 새로운 라그랑주乘數  $\nu$ 를 도입하여 系統方程式과 함께 평가함수에 확장시키면

$$\bar{J} = \phi(\mathbf{X}(N)) + \nu^T \boldsymbol{\psi}(\mathbf{X}(N)) + \sum_{i=0}^{N-1} L^i(\mathbf{X}(i),$$

$$\mathbf{u}(i)) + \boldsymbol{\lambda}^T(i+1) \{ f^i(\mathbf{X}(i), \mathbf{u}(i)) - \mathbf{X}(i+1) \} \quad (19)$$

와 같다. 편의상  $H^i$ 와  $\Phi$ 를 다음과 같이 定義하자

$$H^i = L^i(\mathbf{X}(i), \mathbf{u}(i)) + \boldsymbol{\lambda}^T(i+1) f^i(\mathbf{X}(i), \mathbf{u}(i)) \quad (20)$$

$$\Phi = \phi(\mathbf{X}(N)) + \nu^T \boldsymbol{\psi}(\mathbf{X}(N)) \quad (21)$$

앞에서와 같은 方法으로 변분을 取하여 最適條

件을 求하여 보면

$$\mathbf{X}(i+1) = f^i(\mathbf{X}(i), \mathbf{u}(i)) \quad (22)$$

$$\boldsymbol{\lambda}(i) = \left[ \frac{\partial f^i}{\partial \mathbf{X}(i)} \right]^T \boldsymbol{\lambda}(i+1) + \left[ \frac{\partial L^i}{\partial \mathbf{X}(i)} \right]^T \quad (23)$$

$$\frac{\partial H^i}{\partial \mathbf{u}(i)} = \frac{\partial L^i}{\partial \mathbf{u}(i)} + \boldsymbol{\lambda}^T(i+1) \frac{\partial f^i}{\partial \mathbf{u}(i)} = 0 \quad (24)$$

이때 境界條件은

$$\mathbf{X}(0) \text{ given} \quad (25)$$

$$\boldsymbol{\lambda}^T(N) = \frac{\partial \Phi}{\partial \mathbf{X}(N)} = \frac{\partial \phi}{\partial \mathbf{X}(N)} + \nu^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{X}(N)} \quad (26)$$

따라서 終端條件이 없는 경우와 같은 條件式으로 주어지고 단지  $\boldsymbol{\lambda}(N)$ 의 境界條件만이 다르게 한다. 그러나 여기에서의 問題는  $\nu$ 의 값을 決定하는 方法이 어렵다는 것이다. 따라서  $\nu$ 의 값을 決定하기 위하여 헤스테네스法을 도입하기로 한다.

즉,  $\nu$ 의 값을 逐次的으로 求하기 위하여 다음과 같이 평가함수를 확장하자.

$$\bar{J} = J + \nu^T \boldsymbol{\psi} + \frac{1}{2} \boldsymbol{\psi}^T C \boldsymbol{\psi} \quad (27)$$

$$C > 0$$

式(27)에 대한 最適條件은 단지  $\boldsymbol{\lambda}(N)$ 의 境界條件만이 다음과 같이 바뀐다.

$$\begin{aligned} \boldsymbol{\lambda}(N)^T &= \frac{\partial \phi}{\partial \mathbf{X}(N)} + \nu^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{X}(N)} + C \boldsymbol{\psi}^T \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{X}(N)} \\ &= \frac{\partial \phi}{\partial \mathbf{X}(N)} + [\nu^T + C \boldsymbol{\psi}^T] \frac{\partial \boldsymbol{\psi}}{\partial \mathbf{X}(N)} \end{aligned} \quad (28)$$

式(26), (27)의 決果로부터

$$\nu^{i+1} = \nu^i + C \boldsymbol{\psi} \quad (29)$$

와 같은 관계식이 얻어진다. 그러나 이때 문제가 되는 것은  $C$ 값의 決定이다. 즉  $\boldsymbol{\psi} \rightarrow 0$ 에 따라  $C \rightarrow \infty$ 가 되어야 하는데, 과연 얼마만큼씩  $C$ 값을 증가시킬 것인가 하는 것이다. 만약  $C$ 값을 너무 크게 키워가면 終端點 이탈치가 振動 혹은 發散을 하는 경우가 發生한다. 本 研究에서는 이와같은 問題를 해결하기 위하여 다음과 같은 알고리즘을 도입하였다.

$$C^{i+1} = \begin{cases} 2C^i & \text{if } \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{i+1} > 0 \\ C^i & \text{if } \boldsymbol{\psi} \cdot \boldsymbol{\psi}^{i+1} < 0 \end{cases} \quad (30-a)$$

$$(30-b)$$

式(30-a)는 과소평가된 경우이므로  $C$ 의 값

을 증가시키고 式 (30-b) 는 과대평가된 경우로 C의 값을 감소시킨다. 이와같은 알고리즘의 도입은 振動을 현저히 억제할 수 있었고 또한 發散의 問題도 많이 改善되었다.

그러나 이와같은 方法만으로는 충분히 振動과 發散의 問題가 解決되지 않는다. 式 (29)의 알고리즘은  $\Psi$ 가 충분히 작을 경우에만 거의 수렴이 보장된다. 따라서 매 반복시마다  $\Psi$ 가 너무 크게 되지 않도록 억제를 하여야만 만족할만한 결과를 얻을 수 있다. 따라서 式(19)에  $\Psi$ 에 대한 페널티項을 추가하면 소기의 목적을 달성할 수 있다.

$$\bar{J} = J + \nu^T \Psi + \frac{1}{2} \Psi^T S \Psi \quad (31)$$

$$S > 0$$

이때의 最適條件도 역시 동일하며 단지 境界條件만이 다음과 같이 바뀐다.

$$\lambda^T(N) = \frac{\partial \phi}{\partial \mathbf{x}(N)} + \nu \frac{\partial \Psi}{\partial \mathbf{x}(N)} + S \Psi \frac{\partial \Psi}{\partial \mathbf{x}(N)} \quad (32)$$

이와같은 方法에 의해  $\Psi$ 가 지나치게 커지는 경우를 방지할 수 있고 또한 逐次的으로  $\Psi \rightarrow 0$ 에 손쉽게 수렴을 시킬 수 있었다. 또한 이때에 S값의 변화는 매 반복시마다 10배 정도로 키워줌이 별다른 問題가 없었다.

이때의 알고리즘은 다음과 같다.

- i) C, S가정
  - ii)  $\mathbf{u}^0$ 가정
  - iii)  $\nu = \nu + C \cdot \Psi$   
 $S = 10 \cdot S$
  - iv)  $\min_{\mathbf{u}} \bar{J}$  (플레처 - 포웰법)
  - v)  $\|\Psi\| \leq \epsilon$  計算끝
  - vi)  $C = \begin{cases} 2C & \text{if } \Psi^{i^T} \cdot \Psi^{i+1} > 0 \\ 1/2C & \text{if } \Psi^{i^T} \cdot \Psi^{i+1} < 0 \end{cases}$
- Go To iii)

### 3. 制御變數의 不等號制約條件

式(16), (17), (18)로 주어지는 問題에 制御變數에 對한 不等號制約條件이 存在하는 경우에 대하여 생각하여 보자

$$\mathbf{u}_{\min} \leq \mathbf{u}(i) \leq \mathbf{u}_{\max} \quad (33)$$

위와같은 條件은 다음과 같은 일반형으로 表示할 수 있다.

$$\mathbf{A} \cdot \mathbf{u} \geq \mathbf{b} \quad (34)$$

특히 制御變數 自體에만 위와같이 條件이 주어진 경우에는 行列 A는 매우 간단한 형태로 된다. 즉 각각의 row에 對해서 한 項만이 그 값을 1로 갖고 나머지는 모두 0인 특수한 형태가 된다.

위와같이 不等號制約條件이 있는 問題의 解法에는 일반적으로 라그랑주乘數法, SUMT, 페널티 函數法, 傾斜投影法 등이 使用되고 있다.

本 研究에서는 制約條件函數가 線型函數로 주어지는 경우 매우 편리하게 使用할 수 있는 傾斜投影法을 使用하기로 한다. 그런데 이때에 問題는 傾斜投影法을 使用하여 許容方向벡타(feasible direction vector)를 求하기 위하여는 投影行列(projection matrix)을 복잡하게 구성하여야 하나, 여기서는 投影行列을 매번 구성하지 않고도 곧바로 許容方向벡타를 球하는 알고리즘을 도입하기로 한다.

지금 初期制御벡타  $\mathbf{u}$ 를 모든 不等號制約條件을 만족하도록 進定을 하고, 特히 그중에서 制約條件의 境界面에 걸린 것이 있을 경우에는 그것들만으로 A 行列을 구성하기로 한다. 그러면 이때 行列 A는 對角行列(diagonal matrix)이 된다. 이 對角行列로부터 投影行列을 구성하여 보자. 投影行列을 P 라하면

$$P = I - A^T [AA^T]^{-1} A \quad (35)$$

와 같이 計算된다. 그런데 A가 對角行列이므로 式(35)의 決果는 역시 對角行列이 되며 또한 不等號制約條件에 걸린 곳의 對角成分은 0이 된다.

그리고 나머지 對角成分은 1의 값을 갖게 되는 특수한 형태로 되는데 이 投影行列로 부터 許容方向벡타  $\mathbf{d}$ 를 求하면  $\mathbf{d}$ 는 不等號制約條件에 걸린 變數에 對해서는 0이 되고 나머지 變數에 對해서는 그라디언트값을 갖게 된다.

$$\mathbf{d} = -P \cdot \Delta f \quad (36)$$

그런데 이때 境界面에 걸린 變數中에서 만약 上限値에 걸린 경우 그 點에서의 傾斜方向이 (+)이거나, 또는 下限値에 걸린 경우 그 點에서의 傾斜方向이 (-)일 경우에는 비록 境界面에 걸렸을 지라도 다음에 改善되는 方向이므로 이때에는 許容方向벡타에 그라디언트값을 그대로 유지 시키도록 한다. 다음에 許容方向벡타  $\mathbf{d}$ 를 求하는 알고리즘을 보인다.

$$i) \mathbf{u}(i) = \begin{cases} UMAX & \text{if } \mathbf{u}(i) > \mathbf{u}_{\max} \\ UMIN & \text{if } \mathbf{u}(i) < \mathbf{u}_{\min} \end{cases}$$

ii)  $\frac{\partial J}{\partial \mathbf{u}(i)}$  計算

iii)  $\mathbf{d} = -\frac{\partial J}{\partial \mathbf{u}(i)}$

$d_i = 0$  if  $\mathbf{u}(i) = \mathbf{u}_{\max}, \frac{\partial J}{\partial \mathbf{u}(i)} < 0$

$\mathbf{u}(i) = \mathbf{u}_{\min}, \frac{\partial J}{\partial \mathbf{u}(i)} > 0$

이와같은 方法으로 許容方向벡터를 구성하였으면 이 방향으로 플레처 - 포웰법을 使用하여 평가함수를 最適化하도록 한다. 단 이때 만약 最適化도중 許容方向벡타가 바뀌는 경우가 發生하면 다시 이때 부터 새로운 許容方向벡타를 구성하여 最適化를 수행한다.

이와같은 方法은 기존의 플레처 - 포웰 프로그램을 약간만 수정하면 특별히 傾斜投影法 프로그램을 짜지 않고도 손쉽게 해결 可能하다.

### 3. 모델系統에의 適用

#### 1. 制約條件이 없는 경우

다음과 같은 線型系統에 對해 本 研究의 結果를 適用하였다.

$$J = \frac{1}{2} \sum_{i=0}^{N-1} (\mathbf{x}^T(i) \mathbf{Q} \mathbf{x}(i) + \mathbf{u}^T(i) \mathbf{R} \mathbf{u}(i))$$

$N=10$

$$\mathbf{A} = \begin{bmatrix} 1.1 & -0.1 & 0 \\ 0.1 & 1.1 & -0.2 \\ 0 & -0.2 & 1.0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 1.5 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.0 \end{bmatrix}$$

$$\mathbf{X}(0) = \begin{bmatrix} -0.5 \\ 0 \\ 1.0 \end{bmatrix} \quad \mathbf{X}(N) = \begin{bmatrix} -0.5 \\ 5.0 \\ -1.0 \end{bmatrix}$$

$-2.0 \leq U \leq 2.0$

그리고 S와 C의 初期値는 다음과 같다.

$$\mathbf{S} = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

이때의 수렴특성은 표 1 과 같으며 最終制御벡타 내의 경로는 그림 1 과 같다.

표 1. 制約條件이 없는 경우의 수렴特性  
Table 1. Convergence characteristics without constraints

| 反復回数 | 評價函數(J) | 傾斜 norm (    Hu    ) |
|------|---------|----------------------|
| 1    | 897.8   | 485.8                |
| 2    | 280.6   | 93.9                 |
| 3    | 43.57   | 17.07                |
| 4    | 38.94   | 15.62                |
| 5    | 33.67   | 14.78                |
| 6    | 32.26   | 14.46                |
| 7    | 31.83   | 14.25                |
| 8    | 31.62   | 13.84                |
| 9    | 31.15   | 3.67                 |
| 10   | 31.01   | 3.17                 |
| 11   | 30.99   | 3.15                 |
| 12   | 30.97   | 0.829                |

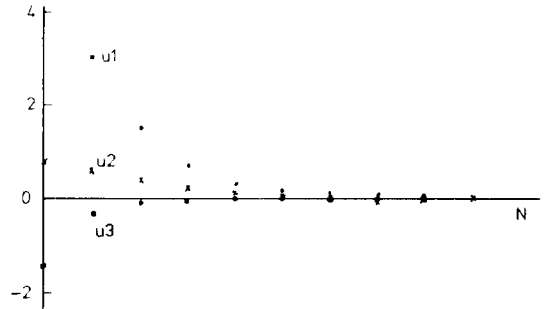


그림 1. 制約條件이 없는 경우의 最適制御  
Fig. 1. Optimal control without constraints

표 1 에서 보듯이 그 수렴속도는 상당히 빠른것을 알 수 있다. 플레처 - 포웰法 대신 傾斜法을 使用할 경우에는 그 수렴속도는 십수배의 差가 있음이 實驗結果나타났다.

#### 2. 終端點에 制約條件이 있는 경우

終端點에 制約條件이 있는 경우에는 헤스테네스法과 페널티函數法을 병용하여 使用하였으며 특히 헤스테네스法の 계수C 값을 앞서 設明한 方法을 도입하여 自動으로 조절하도록 하였다. 이때의 終端

點 離脱値와 수렴과정을 표 2 에 보이고, 또 이때의 最適制御  $\mathbf{u}$  는 그림 2 와 같다.

표 2. 終端條件이 있는 경우의 수렴 特性  
Table 2. Convergence characteristics with terminal constraints

| 反復回数 | 評價函數 ( $J$ ) | 傾斜 norm ( $\ Hu\ $ ) |
|------|--------------|----------------------|
| 1    | 897.8        | 27.31                |
| 2    | 355.7        | 8.73                 |
| 3    | 147.6        | 3.80                 |
| 4    | 168.9        | 0.84                 |
| 5    | 105.6        | 0.89                 |
| 6    | 82.1         | 0.84                 |
| 7    | 79.58        | 0.84                 |
| 8    | 78.58        | 0.84                 |
| 9    | 77.74        | 0.84                 |
| 10   | 77.82        | 0.84                 |
| 15   | 69.33        | 0.09                 |
| 19   | 68.66        | 0.09                 |

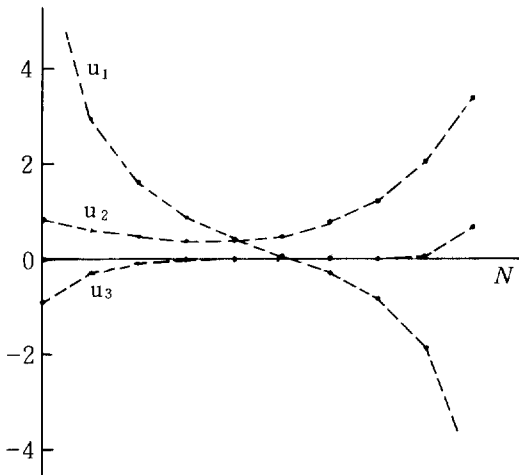


그림 2. 終端條件이 있는 경우의 最適制御  
Fig. 2. Optimal control with terminal constraints

가되 단지 플레처-포웰法을 使用할 경우 變數로서는 傾斜投影벡터만을 取함이 앞서의 경우와 차이가 있다. 즉 매 반복계산 初期에 傾斜投影벡터를 구성하여 만약 이 벡터가 前과 동일하면 그대로 플레처-포웰法의 과정에 따라서 最適化를 수행하고, 만약 變換을 경우에는 다시 처음부터 플레처-포웰法을 使用하도록 한다. 이 경우의 수렴 特性과 最適制御  $\mathbf{u}$  의 경로를 표 3 과 그림 3 에 보인다. 단 이때  $-2.0 \leq \mathbf{u} \leq 2.0$  이다.

표 3. 終端條件과 制御制約이 있는 경우의 수렴 特性

Table 3. Convergence characteristics with terminal constraints and control inequality

| 反復回数 | 評價函數 ( $J$ ) | 終端點離脱値 ( $\ \psi\ $ ) |
|------|--------------|-----------------------|
| 1    | 897.8        | 27.31                 |
| 2    | 355.7        | 8.73                  |
| 3    | 147.6        | 3.8                   |
| 4    | 168.9        | 0.84                  |
| 5    | 168.8        | 0.73                  |
| 10   | 144.8        | 0.32                  |
| 15   | 97.2         | 0.51                  |
| 20   | 102.2        | 0.48                  |
| 25   | 96.95        | 0.54                  |
| 30   | 95.12        | 0.21                  |

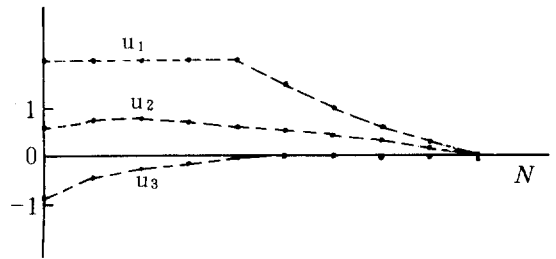


그림 3. 終端條件과 制御制約이 있는 경우의 最適制御

Fig. 3. Optimal control with terminal constraints and control inequality constraints

3. 終端條件과 制御制約이 있는 경우  
이 경우에는 終端條件이 있는 問題로 풀어 나아

4. 結 論  
本 研究는 大型動的系統에서의 最適制御問題를

손쉽게 풀 수 있는 解法에 關한 것으로 實驗 結果 만족할만한 結果가 얻어졌다. 事例研究를 통한 結果, 다음과 같은 結論을 얻었다.

1. 플레처-포웰法의 도입으로 全体的인 計算速度의 向上과 解의 安定성을 얻었다. 이것은 大型動的系統에의 適用이 可能함을 뜻한다.

2. 終端條件이 存在하는 경우에 헤스테네스法과 페널티函數法의 병용으로 終端點에 安定하게 수렴을 시킬 수 있었으며, 특히 헤스테네스法에서의 常數 C의 수정 알고리즘을 開發하여 發散하는問題와 振動의 問題를 크게 改善하였고, 또한 여타의 메모리나 스위프法등에서와 같은 복잡한 計算을 필요하지 않아 大型系統인 경우 그 효과는 더욱 클 것으로 사료된다.

3. 制御變數에 不等號制約條件이 있는 경우에 傾斜投影法과 플레처-포웰法의 병용으로 全体 反復回數를 크게 줄일 수 있었으며, 또한 投影方向 벡타의 決定을 간단히 처리할 수 있는 알고리즘을 開發하여 기존의 플레처-포웰法을 손쉽게 使用할 수 있게 하였으며, 投影行列을 使用하지 않으므로 計算機의 메모리나 計算時間의 상당한 절감을 가져왔고 大型系統일 경우에도 별다른 問題없이 그대로 適用 可能케 하였다.

參 考 文 獻

- 1) Bryson/Ho: "Applied optimal control," Ginn, 1969
- 2) Kirk: "Optimal control theory," Prentice-Hall Hall, 1970
- 3) Hestenes: "Optimization theory: the finite dimensional case," Wiley, 1970
- 4) S.R. McReynolds: "The successive sweep method and dynamic programming," J. Math. Anal. & Appl. vol.3, 1967, pp. 565
- 5) L.S. Lasdon, S. Mitton, and A. Warren: "The method of conjugate gradient for optimal control problems," IEEE Trans. AC, April, 1967
- 6) Okudaira, M.: "A numerical approach to the solution of nonlinear optimal control problems," Ph.d Thesis, march, 1979
- 7) David G. Luenberger: "Introduction to linear and nonlinear programming," Addison-wesley, 1973