

Davidenko법에 의한 시간최적 제어문제의 수치해석해

윤 중 선*

The Numerical Solution of Time-Optimal Control Problems by Davidenko's Method

Joong-sun Yoon*

ABSTRACT

A general procedure for the numerical solution of coupled, nonlinear, differential two-point boundary-value problems, solutions of which are crucial to the controller design, has been developed and demonstrated. A fixed-end-points, free-terminal-time, optimal-control problem, which is derived from Pontryagin's Maximum Principle, is solved by an extension of Davidenko's method, a differential form of Newton's method, for algebraic root finding. By a discretization process like finite differences, the differential equations are converted to a nonlinear algebraic system. Davidenko's method reconverts this into a pseudo-time-dependent set of implicitly coupled ODEs suitable for solution by modern, high-performance solvers. Another important advantage of Davidenko's method related to the time-optimal problem is that the terminal time can be computed by treating this unknown as an additional variable and supplying the Hamiltonian at the terminal time as an additional equation. Davidenko's method was used to produce optimal trajectories of a single-degree-of-freedom problem. This numerical method provides switching times for open-loop control, minimized terminal time and optimal input torque sequences. This numerical technique could easily be adapted to the multi-point boundary-value problems.

Key Words : Davidenko's Method, Pontryagin's Maximum Principle, Time-optimal Control Problems, Multipoint Boundary-value Problems

1. 서 론

최적제어이론, 비선형진동, 응용역학 분야에서 많이

볼 수 있는 어렵지만 아주 중요한 문제들 중에 여러점에서의 값이 주어져 있을 때 비선형미분방정식을 풀어야 하는 경계값 문제가 있다. 비선형 여러점 경계값 문

* 부산대학교 정밀기계공학과, 기계기술연구소

제 (multipoint boundary-value problem)를 해석적으로 풀기가 보통 불가능하므로 수치해석적으로 풀어야만 할 때가 많다.

종래의 방법들은 선형, 낮은 차수, 스칼라 입력, 고정된 끝남시간, 이동 끝점 문제 (linear, low-order, scalar input, fixed-terminal-time, and/or moving-end-point problems)와 같은 제한된 문제들을 다루고 있다. 이 방법들의 성공 여부는 문제의 특성과 좋은 초기값의 가정에 달려 있다. 따라서 비선형 높은 차수 문제를 풀, 구현하기 쉽고 체계적인 방법이 필요하다. 이 논문에서는 대수계의 근을 찾는 방법의 하나인 Davidenko법을 따른 새로운 수치해석법을 일반적인 여러점 경계값 문제를 풀 방법으로 제안하고자 한다. 이 방법은 한정된 입력, 고정된 끝점, 자유로운 끝남시간 (a bounded input, fixed-end-points and free-terminal-time problem)을 가지는 시간최적 제어문제를 정의하는 2점 경계값 문제 (two-point boundary-value problem)에 적용될 수 있다.

유한차분 (finite difference)법과 같은 이산화 (discretization) 과정을 쓰면 미분방정식들을 많은 수의 비선형 대수방정식의 제로 바꿀 수 있다. 대수방정식의 근을 찾는 알고리즘의 하나인 Davidenko법^{1, 2, 3}에 따라 새로 세워진 상미분방정식의 계는 적절한 상미분방정식 풀이 프로그램을 써서 수치해석적으로 풀 수 있다. 이렇게 하여 Davidenko법은 2점 경계값 문제에 쓰일 수 있다. 이러한 2점 경계값 문제의 풀이법은 여러점 경계값 문제에 바로 쓰일 수 있다.

이 논문에서는 2점 경계값 문제를 풀기 위해 Davidenko법을 따른 하나의 수치해석적 풀이 과정을 제안하고자 한다. 한정입력을 가지는 시간최적 제어문제의 답을 줄 필요조건을 먼저 끌어내고 이러한 고정된 끝점, 자유로운 끝남시간을 가지는 시간최적 제어기의 설계 문제는 비연속 입력을 가지는 2점 경계값 문제로 바꾸어 나타낼 수 있다. 다음으로 Davidenko법을 2점 경계값 미분계에 적용한다. 구동진자의 예가 제안된 방법을 보여주기 위해 쓰인다. 마지막으로 문제의 형성과 결과에 대한 자세한 토론이 포함된다.

2. 2점 경계값 시간최적 제어문제

고정된 끝점과 자유로운 끝남시간 (fixed-end-points and free-terminal-time)을 가지는 시간최적 제어문

제는 다음처럼 나타내어진다. x 가 상태벡터, $x = (x_1, x_2, \dots, x_n)^T$ 이고 u 는 제어입력벡터, $u = (u_1, u_2, \dots, u_r)^T$ 일 때 초기상태 x_0 , 종료상태 x_1 과 제한조건을 만족하는 제어 조합 U 를 가지는 연속동적계

$$\dot{x}(t) = f(x(t), u(t), t) \tag{1a}$$

$$x(t_0) = x_0 \tag{1b}$$

$$x(t_1) = x_1 \tag{1c}$$

$$u(t) \in U \text{ 모든 } t \text{에서} \tag{1d}$$

를 x_0 에서 x_1 로 최단시간에 옮기는 U 에 속하는 제어기를 설계하라. 평가함수 J 는

$$J = \int_{t_0}^{t_1} f_0(x(t), u(t)) dt \tag{1e}$$

이다.

최적해의 풀이는 Pontryagin의 최대정리 (Maximum Principle)를 써서 끌어낼 수 있다.⁴ Hamiltonian

$$H(x, \varphi, u, t) = \sum_{i=1}^n \varphi_i f_i(x, u) \tag{2}$$

에서 상호상태 (covariant) 방정식과 상태방정식의 표준계

$$\dot{\varphi}_i = -\frac{\partial H}{\partial x_i}, \quad i = 1, 2, \dots, n, \tag{3}$$

$$\dot{x}_i = \frac{\partial H}{\partial \varphi_i}, \quad i = 1, 2, \dots, n. \tag{4}$$

를 만든다. 최적제어는 x_0 을 x_1 로 움직이는데 드는 전 시간에 걸쳐 Hamiltonian을 최적화하는 제어이다. 최적제어 전략이 실패를 거두려면 서로 얽힌 (coupled) 상태방정식과 상호상태방정식을 풀어야 한다. 이는 시간 t_0 와 t_1 에서의 경계값 x 를 가지는 변수 x 와 φ 의 $2n$ 개의 비선형 1차 상미분방정식의 조합 즉 2점 경계값 문제가 된다. 많은 수의 방정식을 가지는 2점 경계값 문제의 정확한 답을 찾는 계산은 매우 어렵다.

2차 성능지수와 특정의 2차나 3차 선형계의 최적을 다루는 문제 부류를 빼고는 해석적으로 답을 찾는 것은 일반적으로 불가능하다. 따라서 최적제어 문제는 반복적인 계산과정을 써서 풀어야 한다.^{5, 6} 특히 최적제어 문제를 위한 여러 수치해석 방법들^{5, 6}이 제안되어 왔다. 한정입력을 가지는 최적제어 문제가 특히 로봇 머니플레이터의 제어기 설계분야에서 많은 관심을 끌고 있다.^{7, 8, 9, 10}

근사시간 최적제어 (near-time-optimal control)는 문에서⁷ 토크 입력의 스위칭시간을 예측하는 수치해석법을 제안했다. 선형 풀린 (decouple) 계에 대해서 최

대원리에 따른 최적제어가 연구되었다. 토크에 대한 뱀 뱀 최적해가 설명되었고 2점 경계값 문제의 수치해석 문제는 시행착오법으로 풀었다. 상호상대변수의 첫 어 림짐작값이 선형화된 계에서 택해지고 이 값을 가지고 대화형 그래픽 단말기의 도움을 받아 더 좋은 값으로 바꾸어 간다. 문제의 답은 이 값으로 미분계를 시간에 따라 역으로 적분해서 얻었다. 그러나 선형화때문에 계 의 전역최적거동(global optimal behavior)은 보장되 지 않았다.

비연속입력을 가지는 문제의 경우 전통적인 기율기탐 사법은 더 이상 쓰일 수 없으므로 adjustable control-variation weight⁽⁶⁾라는 새로운 기율기 알고리즘 이 최단시간에 주어진 거리를 가는 제한된 입력을 가지는 제어문제에 대해 제안되었다.

고정시간, 이동끝점(fixed-time, moving-end-point) 풀의 최적제어문제⁽¹⁰⁾는 initial value adjusting법⁽¹¹⁾을 써서 Pontryagin의 최대정리에 따라 연구 되었다. 이 방법에서 미분방정식은 알지 못하는 첫 짐 작값을 가지고 상미분 초기값 문제로 풀 수 있다. 이전 의 반복에서 얻은 경계조건 값과 이전의 초기조건을 조 금 바꾸어 얻은 방정식의 경계조건값의 차이에서 빠진 초기조건값을 수치해석적으로 계산할 수 있게 한다. 이 러한 종류의 방법은 많은 계산을 필요로 하고 traversality 조건에서 추가의 경계에 대한 정보를 주는 고정 시간, 이동끝점 문제와 같은 제한된 계에 적합하다.

Geering⁽⁹⁾은 한정입력을 가지는 2 자유도 머니폴레 이터의 시간최적 풀이법을 제시하였다. 문제의 계산시 간을 줄이고 미리 주어진 마지막 상태에 대한 페널티법 을 써서 2점 경계값 문제를 푸는 슈팅 알고리즘과 파라 미터 최적화 법이 다양한 파라미터화 제어에 대한 스위 칭시간을 찾는데 쓰였다. 슈팅 알고리즘의 수렴은 마지 막 상호상태 벡터의 첫 짐작값을 이전 계산단계의 페널 티 항의 Lagrange 곱을 써서 달성되었다.

이러한 방법들의 성공여부는 문제의 특성에 달려 있 고 좋은 초기 짐작값을 필요로 한다. 대수 문제의 근을 찾는 방법의 하나인 Davidenko법에 따른 새로운 수치 해석법을 앞의 2점 경계값 문제를 풀기 위해서 다음절 에서 설명하도록 한다.

3. Davidenko법의 시간최적 제어문제에의 응용

비선형 대수방정식으로 나타내어지는 계

$$\underline{f}(\underline{x}) = 0 \tag{5}$$

의 근 $\underline{x} = (x_1, x_2, \dots, x_n)^T$ 을 구하는 방법인 Davi denko법^(3, 12)의 기본 꼴은 τ 가 인위적 독립변수일 때 내재적(implicit) 꼴

$$\underline{J} \frac{d\underline{x}}{d\underline{\tau}} = -\underline{f}(\underline{x}) \tag{6}$$

또는 외현적(explicit) 꼴

$$\frac{d\underline{x}}{d\underline{\tau}} = -\underline{J}^{-1} \underline{f}(\underline{x}) \tag{7}$$

로 나타낼 수 있다. 여기서 자코비안 행렬 \underline{J} 는

$$\underline{J} = \begin{bmatrix} \partial f_1 / \partial x_1 & \partial f_1 / \partial x_2 & \dots & \partial f_1 / \partial x_n \\ \partial f_2 / \partial x_1 & \partial f_2 / \partial x_2 & \dots & \partial f_2 / \partial x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial f_n / \partial x_1 & \partial f_n / \partial x_2 & \dots & \partial f_n / \partial x_n \end{bmatrix} \tag{8}$$

이다. 초기조건 벡터 $\underline{x}(0)$ 를 가지는 n 개의 초기값 상 미분방정식계는 정상상태 $d\underline{x}/d\underline{\tau} = 0$ 즉 $\underline{f} = 0$ 에 이 를 때까지 수치해석적으로 적분을 해서 답을 얻을 수 있다.

n 이 작으면 자코비안의 역은 해석적으로 또는 수치해 석적으로 찾을 수 있고 외현적 초기값 상미분방정식 풀 이 프로그램인 DSS/2,⁽¹³⁾ LSODE,⁽¹⁴⁾ DGEAR⁽¹⁵⁾등이 쓰일 수 있다. 역 자코비안을 찾기가 매우 어렵거나 불 가능할 때에는 뛰어난 내재적(implicit) 초기값 상미분 방정식 풀이 프로그램인 LSODI⁽¹⁴⁾나 DASSL⁽¹⁶⁾이 내 재적 계인 식(6)에 바로 쓰일 수 있다.

이 절에서는 2절에서 기술된 시간최적 문제인 2점 경 계값 미분계인 식 (3)과 (4)에 Davidenko법을 쓰려 한다. 유한차등법과 같은 이산과정을 써서 미분계를 대 수계로 바꾼다.^(13, 17, 18)

2절의 문제에 대해 새로운 종속변수 벡터 $\underline{y}^T = [\underline{x}^T, \underline{\varphi}^T]$ 를 생각하자. 이때 \underline{x} 는 $\underline{x} = (x_1, x_2, \dots, x_n)$ 상태벡 터이고 $\underline{\varphi}$ 는 $\underline{\varphi} = (\varphi_1, \varphi_2, \dots, \varphi_n)^T$ 상호상태벡터이다. 상태방정식과 상호상태방정식 (3)과 (4)는 $2n$ 개의 1차 미분방정식의 조합

$$\frac{dy_i}{dt} = f_i(\underline{y}, \underline{u}, t), \quad i = 1, 2, \dots, 2n, \tag{9}$$

으로 쓸 수 있다. 여기서 \underline{u} 는 $\underline{u} = (u_1, u_2, \dots, u_n)^T$ 입력

벡터이고 엇갈림(split) 경계조건

$$y_i(t_0) = x_i(t_0), \quad y_i(t_1) = x_i(t_1), \quad i = 1, 2, \dots, n. \quad (10)$$

을 가진다.

식 (9)가 구간일정(fixed stepsize) 알고리즘을 써서 수치 적분을 한다면 답 $y_i(t)$ 는 시간축에서 $M+1$ 간격이 같은 이산점에서 존재한다. 여기서 $M+1$ 은 각 y_i 에 대한 경계점을 포함하여 시간 영역에서의 격자(grid) 점들의 수이다. 연속값 $y_i(t)$ 를 다음과 같이 비연속 값 q_i 로 바꾼다.

$$q_i = (q_i^0, q_i^1, \dots, q_i^M)^T \quad (11)$$

여기서 $q_i^j = y_i(t_0 + (j)\Delta T)$ 이고 $\Delta T = (t_1 - t_0)/M$ 이며 $j = 0, 1, \dots, M$ 이다. 식 (9)의 이산시간에서의 미분 dy_i/dt 는 유한차분법 같은 것을 쓰면 답들의 점에서 q_i^j 의 선형조합으로 비슷하게 나타낼 수 있다.

그러면 식 (9)는 이산적인 꼴

$$A_i q_i - \underline{f}_i(q_1, q_2, \dots, q_{2n}, u_1, u_2, \dots, u_r) = 0, \quad i = 1, 2, \dots, 2n \quad (12a)$$

로 정리될 수 있다. 여기서 A_i 는 1차 미분계수를 위한 $(M+1) \times (M+1)$ 계수 행렬이고 $u_k(t)$ 는 $M+1$ 격자점에서의 이산벡터 $u_k = (u_k^0, u_k^1, \dots, u_k^M)^T$ 이다. 또는 $2n(M+1)$ 벡터 꼴로 나타내면 식 (12a)는

$$F(Q, U) = A Q - \underline{f} = 0 \quad (12b)$$

가 된다. 여기서 Q 는 이산화된 $2n(M+1)$ 개의 상태 벡터 $Q = (q_1^T, q_2^T, \dots, q_{2n}^T)^T$ 이고 \underline{f} 는 이산화된 $2n(M+1)$ 개의 벡터 $\underline{f} = (f_1^T, f_2^T, \dots, f_{2n}^T)^T$ 이고 U 는 이산화된 $r(M+1)$ 개의 입력 벡터 $U = (u_1^T, u_2^T, \dots, u_r^T)^T$ 이다. 여기서 F 는 $2n(M+1)$ 크기의 벡터이고 A 는 $2n(M+1) \times 2n(M+1)$ 크기의 행렬은

$$A = \begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_{2n} \end{bmatrix} \quad (13)$$

이다.

예를 들면 $y_i(t)$ 가 같은 간격일 때 3점 Lagrange 다항식⁽¹³⁾

$$y_i(t) = \frac{(t-t^j)(t-t^{j+1})}{(t^{j-1}-t^j)(t^{j-1}-t^{j+1})} q_i^{j-1} + \frac{(t-t^{j-1})(t-t^{j+1})}{(t^j-t^{j-1})(t^j-t^{j+1})} q_i^j + \frac{(t-t^{j-1})(t-t^j)}{(t^{j+1}-t^{j-1})(t^{j+1}-t^j)} q_i^{j+1} \quad (14)$$

로 근사되면 이 3점 미분 공식은

$$\begin{bmatrix} \dot{y}_i^{j-1} \\ \dot{y}_i^j \\ \dot{y}_i^{j+1} \end{bmatrix} = \frac{1}{\Delta T} \frac{1}{2!} \begin{bmatrix} -3 & 4 & -1 \\ -1 & 0 & 1 \\ 1 & -4 & 3 \end{bmatrix} \begin{bmatrix} q_i^{j-1} \\ q_i^j \\ q_i^{j+1} \end{bmatrix} \quad (15)$$

으로 요약될 수 있다. 행렬의 첫번째 행은 $j=1$ 일 때인 \dot{y}_i^0 을 위해 쓰일 때이고 두번째 행은 $1 \leq j \leq M-1$ 일 때인 \dot{y}_i^j 을 위해 쓰일 때이며 세번째 행은 $j=M-1$ 일 때인 \dot{y}_i^M 을 위해 쓰일 때이다. 행렬 A_i 는 Bickley 행렬⁽¹³⁾로 Bickley 행렬 변수 n, m, p 가 $n=2, m=1, p=0, 1, 2$ 일 때이다. 미분계수에 대해서 간격이 같지 않거나 차수가 높은 다른 근사 방법을 쓸 수도 있다. 행렬 A_i 는 보통 띠틈대칭 꼴을 가진다. 식 (15)의 $(M+1) \times (M+1)$ 크기의 행렬 A_i 는 다음과 같다.

$$A_i = \frac{1}{\Delta T} \frac{1}{2!} \begin{bmatrix} -3 & 4 & -1 & \cdot & \cdot & \cdot \\ -1 & 0 & 1 & \cdot & -0 & \cdot \\ 0 & -1 & 0 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & -0 & \cdot & -1 & 0 & 1 \\ \cdot & \cdot & \cdot & 1 & -4 & 3 \end{bmatrix} \quad (16)$$

Davidenko법을 쓴 대수계 (12)는 인위적 독립변수 τ 의 항을 가지는

$$J \frac{dQ}{d\tau} = -F \quad (17)$$

로 바뀌어 나타내어 진다. 식 (17)은 선형 내재적 초기 값 상미분계로 된 문제의 엇갈림 경계값인 식 (10)은 아직 고려되지 않았다. 경계값을 포함시키는 방법은 다음에 예제와 함께 설명될 것이다.

지금까지의 방법은 시간과 공간 차원의 편미분방정식 조합의 답의 근사화 과정인 선수치법(numerical method of lines, NMOL)을 많이 닮아 있다. 원 독립변수인 시간 t 를 간격이 같도록 이산화시키고 시간에

대한 미분계수는 좌우대칭 유한차분(centered finite-difference) 근사로 바꾸어질 수 있다. 시간은 선수치법(NMOL)에 따른 수식 근사화에서의 공간 독립변수처럼 생각할 수 있다. 그러면 시간같은 독립변수 τ 에 대해서 미분방정식을 가변간격(variable-step), 가변차수(variable order)의 범용 수치해석 프로그램들을 써서 수치해석적으로 적분할 수 있다.

Edelen²⁾은 Davidenko법과 같은 미분 과정이 효과적인 수치해를 줄 이러한 방정식 계의 재구성 과정에서 문제를 지적하고 있다. 불행하게도 시간최적 문제의 경우 자코비안 J 는 항상 특이 행렬이고 Bickley 행렬의 함으로 상대방정식과 상호상대방정식의 함의 자코비안은 보통 매우 성기다. (sparse)

자코비안 J 는 많은 경우 특이 성질을 가지거나 조건이 좋지 않으므로(ill-conditioned) 내재적 프로그램이 주로 쓰인다. 수렴영역은 모든 한정상태로 이루어지거나 계산 과정에서 가상(spurious) 평형을 만들어 낼 수 있다. 결과를 원 식에 넣어 보거나 구한 제어기를 가지는 제어계의 시뮬레이션을 해보아서 얻은 답의 유효함을 따져볼 수 있다.

4. 1 자유도 예제

2점 경계값 문제인 시간최적 제어문제에 Davidenko법을 쓴 예로 구동 진자(driven pendulum)를 살펴 보자.

구동 진자의 운동 방정식은

$$\frac{d^2\theta}{dt^2} = \frac{\tau_0}{ml^2} - \frac{g}{l} \sin\theta \quad (18)$$

이고 여기서 θ 는 각도, m 은 질점, l 은 진자의 길이, τ_0 는 가해진 토크이다. 해당 2점 경계값 문제는

$$\underline{y} = (\theta_1, \theta_2, \varphi_1, \varphi_2)^T \quad (19a)$$

$$\begin{aligned} \dot{y}_1 &= f_1 = y_2 & y_1(0) &= \theta_1 \\ \dot{y}_2 &= f_2 = a_1 \tau^* - a_2 \sin y_1 & y_1(t_f) &= \theta_f \\ \dot{y}_3 &= f_3 = a_2 y_4 \cos y_1 & y_2(0) &= 0 \\ \dot{y}_4 &= f_4 = -y + 3 & y_2(t_f) &= 0 \\ \tau^* &= \tau_{\max} \text{sign}(\varphi_2) \end{aligned} \quad (19b)$$

로 쓸 수 있고 여기서 $a_1 = 1/ml^2$, $a_2 = g/l$, $\theta_1 = \theta$, $\theta_2 = \theta$ 이다.

3절에서 보듯 모든 시간 미분 dy_i/dt 을 유한차등 근사로 바꿀 수 있도록 시간축에서 같은 간격, $\Delta T = t_f/M$ 이 쓰인다. 각 $M+1$ 이산시간 격자점에서 $2n$ ($n=2$)개의 대수 방정식으로 나타내어 진다. 식 (19)의 이산시간 식은

$$j = 0 \text{ 일 때} \\ F_i^0 = 0 = (-3y_i^0 + 4y_i^0 - y_i^2)/2\Delta T - f_i^0 \quad (20a)$$

$$j = 1, 2, \dots, M-1 \text{ 일 때} \\ F_i^j = 0 = (-y_i^{j-1} + y_i^{j+1})/2\Delta T - f_i^j \quad (20j)$$

$$j = M \text{ 일 때} \\ F_i^M = 0 = (y_i^{M-2} - 4y_i^{M-1} + 3y_i^M)/2\Delta T - f_i^M \quad (20M)$$

또는 벡터 꼴로는

$$\underline{F}(y_1, y_2, y_3, y_4, \underline{\tau}) = 0 \quad (21)$$

으로 \underline{F} 는 $4(M+1)$ 차원 벡터, \underline{y}_i 와 $\underline{\tau}$ 는 모두 $(M+1)$ 차원 벡터이다. 해당 경계조건은

$$\begin{aligned} y_1^0 &= \theta_1 & y_1^M &= \theta_f \\ y_2^0 &= 0 & y_2^M &= 0 \end{aligned} \quad (22)$$

이다. 이 4개의 경계 값들에 대한 방정식은 대수계인 식 (21)에서 추려져서 \underline{F} 에 대한 $4(M)$ 개의 방정식이 된다. 자세한 과정은 다음 절에서 설명하고자 한다.

그러면 구동 진자에 대한 초기값 상미분방정식계는 식 (6)에서와 같이

$$\underline{J}(\underline{y}) = \frac{d\underline{y}}{d\underline{\tau}} = -\underline{F} \quad (23)$$

로 나타내어 진다.

5. 진자예제의 수치결과

이 절에서는 구현 기술과 구현시 어려운 점 몇가지를 진자 예로 설명하고자 한다. 경계조건인 식 (22)를 대수계인 식 (20)에 수렴을 좋게 하는 쪽으로 짊어 넣으려고 한다.

경계값에 대한 상대방정식의 이산식 (20a)와 (20M)은 위치변수 y_i ($i=1$)에 대해

$$F_1^0 = 0 = (-3y_1^0 + 4y_1^1 - y_1^2)/2\Delta T - y_2^0 \quad (24a)$$

$$F_1^M = 0 = (y_1^{M-2} - 4y_i^{M-1} + 3y_i^M) / 2\Delta T - y_2^M \quad (24b)$$

이다. 여기서 경계값들은

$$y_1^0 = \theta_i \quad y_1^M = \theta_f \quad (24c)$$

이다. 속도변수 y_2 ($i=2$)에 대해

$$F_2^0 = 0 = (-3y_2^0 + 4y_2^1 - y_2^2) / 2\Delta T - (a_1\tau^{*0} - a_2 \sin y_1^0) \quad (24d)$$

$$F_2^M = 0 = (y_2^{M-2} - 4y_2^{M-1} + 3y_2^M) / 2\Delta T - (a_1\tau^{*M} - a_2 \sin y_1^M) \quad (24e)$$

이다. 여기서 경계값은

$$y_2^0 = 0 \quad y_2^M = 0 \quad (24f)$$

이다. 식 (24a), (24b), (24d), (24e)에 경계조건식 (24c)와 (24f)를 집어 넣을 수 있다. 경계조건을 대수계의 식에 바로 넣으면 프로그램은 아직 답에 이르지 않았는데도 계산간격을 너무 줄여버려 답을 주지 못하고 컴퓨터 시간만 오래도록 써버린다.

이러한 문제에서 벗어나려면 경계값에 달하는 행과 열을 자코비안 행렬에서 추려서 4Mx4M 크기의 행렬을 만든다. 경계조건을 식에 넣으면 자코비안 행렬 모습의 부적절한 입력 때문에 톱날 풀의 답을 가지게 되었다. 자코비안 행렬의 이러한 성질을 없앨 경계값 근처에서 식을 다시 세우는 방법을 설명하도록 한다.

풀어야 할 방정식의 수는 4(M+1)개에서 진자의 위치와 속도에 대한 경계값에 대한 4개의 방정식인 식 (24)를 추려내어서 4M개로 줄어든다. 단지 이러한 방정식을 버리기보다는 $j=1, 2, M-2, M-1$ 에서의 인접 격자에서의 변수에의 제한조건으로 다시 쓰이도록 한다.

진자의 이산위치변수 y_1^j 와 y_2^j 는 경계값에 대한 방정식 $F_1^0 = 0$ 으로부터 다음과 같이 계산된다.

$$y_1^1 = (3y_1^0 + y_1^2 + 2\Delta T y_2^0) / 4 \quad (25a)$$

$$y_2^1 = (-3y_1^0 + 4y_1^1 - 2\Delta T y_2^0) \quad (25b)$$

여기서 $y_1^0 = \theta_i$ 와 $y_2^0 = 0$ 이고 위치변수 y_i^{M-2} 와 y_i^{M-1} 는 경계값에 대한 방정식 $F_1^0 = 0$ 으로부터 다음과 같이 계산된다.

$$y_1^{M-2} = (4y_1^{M-1} - 3y_1^M + 2\Delta T y_2^M) \quad (25c)$$

$$y_1^{M-1} = (y_1^{M-2} + 3y_1^M - 2\Delta T y_2^M) / 4 \quad (25d)$$

여기서 $y_1^M = \theta_f$ 이고 $y_2^M = 0$ 이다. 마찬가지로 진자 속도의 이산변수 $y_2^1, y_2^2, y_2^{M-2}, y_2^{M-1}$ 는 경계값에 대한 방정식 $F_2^0 = 0$ 와 $F_2^M = 0$ 으로부터 다음과 같이 계산된다.

$$y_2^1 = (3y_2^0 + y_2^2 + 2\Delta T(a_1\tau^{*0} - a_2 \sin y_1^0)) / 4 \quad (26a)$$

$$y_2^2 = (-3y_2^0 + 4y_2^1 - 2\Delta T(a_1\tau^{*0} - a_2 \sin y_1^0)) \quad (26b)$$

$$y_2^{M-2} = (4y_2^{M-1} - 3y_2^M + 2\Delta T(a_1\tau^{*M} - a_2 \sin y_1^M)) \quad (26c)$$

$$y_2^{M-1} = (y_2^{M-2} + 3y_2^M - 2\Delta T(a_1\tau^{*M} - a_2 \sin y_1^M)) / 4 \quad (26d)$$

여기서 $\tau^{*0} = \tau_{\max} \text{sign}(y_4^0)$ 이고 $\tau^{*M} = \tau_{\max} \text{sign}(y_4^M)$ 이다.

이러한 변수들인 식 (25)와 (26)은 진자의 위치와 속도($i=1, 2$)의 대수제인 식 (20)은 다음과 같이 쓸 수 있다.

$$j=1; \quad F_i^1 = 0 = (-y_i^0 + y_i^2) / 2\Delta T - f_i^1 \quad (27a)$$

$$j=2; \quad y_i^1 = (3y_i^0 + y_i^2 + 2\Delta T f_i^0) / 4 \\ F_i^2 = 0 = (-y_i^1 + y_i^3) / 2\Delta T - f_i^2 \quad (27b)$$

$$j=3; \quad y_i^2 = (-3y_i^0 + 4y_i^1 - 2\Delta T f_i^0) \\ F_i^3 = 0 = (-y_i^2 + y_i^4) / 2\Delta T - f_i^3 \quad (27c)$$

$$j=4, 5, \dots, M-4; \quad F_i^j = 0 = (-y_i^{j-1} + y_i^{j+1}) / 2\Delta T - f_i^j \quad (27d)$$

$$j=M-3; \quad y_i^{M-2} = (4y_i^{M-1} - 3y_i^M + 2\Delta T f_i^M) \\ F_i^{M-3} = 0 = (-y_i^{M-4} + y_i^{M-2}) / 2\Delta T - f_i^j \quad (27e)$$

$$j=M-2; \quad y_i^{M-1} = (y_i^{M-2} + 3y_i^M - 2\Delta T f_i^M) / 4 \\ F_i^{M-2} = 0 = (-y_i^{M-3} + y_i^{M-1}) / 2\Delta T - f_i^j \quad (27f)$$

$$j=M-1; \quad F_i^{M-1} = 0 = (-y_i^{M-2} + y_i^M) / 2\Delta T - f_i^j \quad (27g)$$

상호상태변수 y_3, y_4 에 대한 대수계는 식 (20)의 원 모습으로 남아 있다. 식 (27)의 이산상태방정식의 수는 2(M+1)개에서 2(M-1)개로 줄어 든다. 반면에 상호상태 쪽의 방정식의 수는 여전히 2(M+1)개이므로 결과로는 4M개의 상태와 상호상태 이산대수계를 가진다. 격자점 11개 즉 M=10일 때 1차 미분의 유한차등 근사의

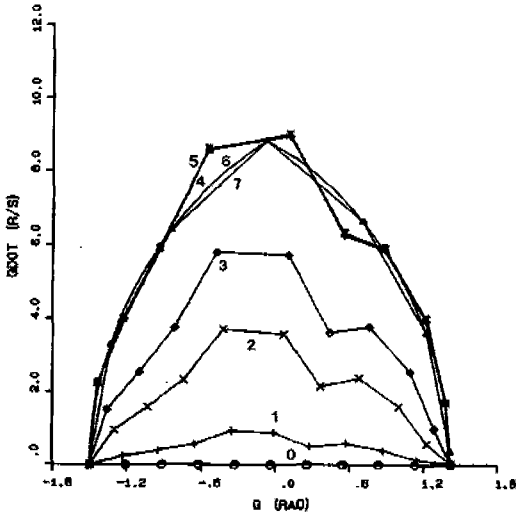


Fig. 1 Time-Optimal Phase Plane Trajectory

이다. 11개의 이산시간에서의 답들만이 계산되므로 스위칭시간의 구간만이 구해진다. 더 잘게 자른 격자간격 즉 21개의 격자점 ($M = 20$)을 쓰면 계산 양은 늘지만 더 좋은 답을 얻을 수 있다. 상태변수와 상호상태변수는 정규화된 시간 (t/t_f)축에서 이산순간일 때를 Fig. 2에 보여준다. 정확한 토크 순서와 수치해석적으로 계산한 토크 순서는 Fig. 3에서 보여준다. Fig. 2와 Fig. 3에서 0은 초기값, 1은 수치해석값, 2는 정확한 값을

나타낸다.

문제의 수렴은 $\tau = 0.1, 0.5, 1.0, 5.0, 10$ 에서의 오차의 제곱합의 제곱근의 자연대수값, $\ln(\sum e_i(\tau) * e_i(\tau))$ 으로 Fig. 4에서 보여 준다. 식 (23)의 계의 오차와 Hamiltonian은 $\tau = 5$ 이상일 때 납득할 만한 값에 다달았다. 모든 수치 해석적 계산은 CDC CYBER 180-850 컴퓨터에서 이루어졌고 단계당 CPU 시간은 약 343 milliseconds이었다. 단계에 따른 자코비안 계산과 나머지 계산은 Table 2에 주어져 있다. 독립 변수에 따른 오차의 지수적 감소를 보여 주고 있다. 수직축은 오차 제곱들 합을 제곱근의 자연 대수를 나타낸다. $\tau = 0.1, 0.5, 1.0, 5.0, 10.0$ 일 때의 오차가 그려져 있다. 오차 $e(\tau)$ 는 τ 에 따라 지수적으로 감소한다.

프로그램에 무관하게 계 (23)의 자코비안 $J_{ij} = \partial F_i / \partial y_j$ 은 종속변수 y_j 를 조금 바꾸었을 때의 F_i 의 바뀜을 계산해서 근사적으로 얻을 수 있다.⁽²⁰⁾

$$\frac{\partial F_i}{\partial y_j} = \lim_{\Delta y_j \rightarrow 0} \left(\frac{\Delta F_i}{\Delta y_j} \right) \quad (30)$$

자코비안 행렬의 모든 요소는 계산되고 Table 3의 근거에 따른 행렬의 각 요소의 크기 차수를 나타낸 행렬이 만들어졌다. 계산된 자코비안 지도는 Fig. 5와 같이 거의 덩어리 띠 모습을 보여 준다.

Table 2 Computational Results for a Pendulum Example on CYBER 180-850

τ	0.0	0.1	0.5	1.0	5.0	10.0
NS		12	20	27	50	65
NJE		4	5	6	8	10
NRE		170	219	266	370	465
NQ		3	4	5	5	5
CPU(sec)		8.025	2.364	2.263	5.086	4.580
$\sum(F^2)$	6.30×10^3	5.12×10^3	2.25×10^3	8.18×10^2	3.78×10^1	5.39×10^0
$\sqrt{\sum(i)}$	7.94×10	7.15×10	4.74×10	2.86×10	6.15×10^1	7.34×10^3
$\ln(\sqrt{\sum(i)})$	4.37	4.27	3.86	3.35	-0.486	-4.91
$\sum(H^2)$	3.99	3.86	2.69	1.31	2.36×10^2	2.40×10^2
$\sqrt{\sum(i)}$	2.00	1.96	1.64	1.15	1.54×10^1	1.55×10^1

CPU Time for JMAP Computation = 0.137 sec
 Error 10^{-6} (Both relative and absolute error)
 M Number of Grid Points(0 to $M = 10$)
 NS Number of Iterations
 NJE Number of Jacobian Evaluations

NRE Number of Residual Evaluations
 NQ Order of BDF
 $\sum F(i)^2$ Summation of Error Square Terms
 $\sum H(i)^2$ Summation of Hamiltonian Square Terms

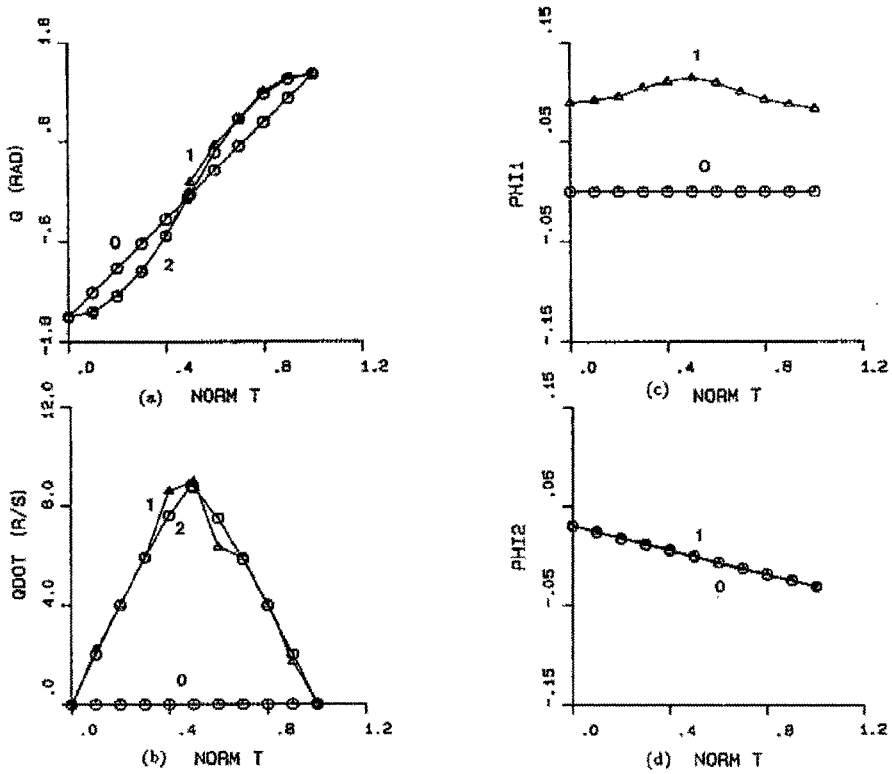


Fig. 2 Time-Optimal State and Covariant Variables VS. Normalized Time

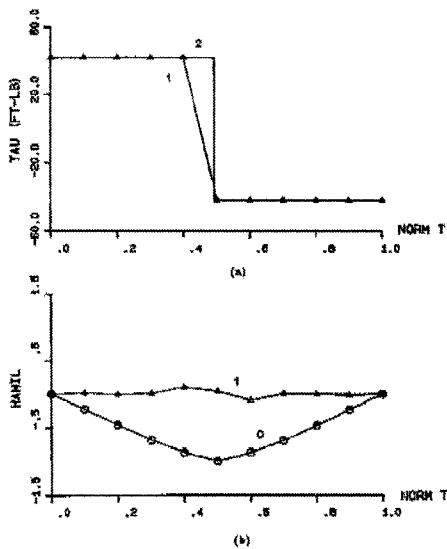


Fig. 3 Time-Optimal Torque Sequences and the Hamiltonian VS. Normalized Time

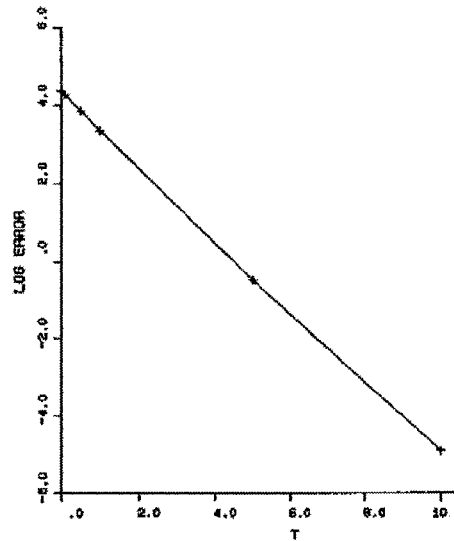


Fig. 4 Error Convergence of the Two-p...

Table 3 Definition of Order-of-magnitude Numbers Used in the Jacobian Maps

$J_{ij} \leq 10^{-5}$	blank	is printed
$10^{-5} < J_{ij} \leq 10^{-4}$	0	is printed
$10^{-4} < J_{ij} \leq 10^{-3}$	1	is printed
$10^{-3} < J_{ij} \leq 10^{-2}$	2	is printed
$10^{-2} < J_{ij} \leq 10^{-1}$	3	is printed
$10^{-1} < J_{ij} \leq 1$	4	is printed
$1 < J_{ij} \leq 10$	5	is printed
$10 < J_{ij} \leq 10^2$	6	is printed
$10^2 < J_{ij} \leq 10^3$	7	is printed
$10^3 < J_{ij} \leq 10^4$	8	is printed
$10^4 < J_{ij}$	9	is printed

6. 결론

비선형 대수방정식 계를 푸는 Davidenko법은 경계 조건이 엇갈리고 끝냄시간을 모르는 미분계를 근사적 풀이에 쓰일 수 있다. 근사과정은 편미분방정식을 상미분방정식의 조합으로 나타내는 선수치법(NMOL)을 많이 닮아 있다. 이 방법은 Pontryagin의 최대원리에서 바로 끌어낸 시간최적 제어계를 푸는데 특히 쓰임새가 있다. 이 문제에서 상호상대계의 경계값과 끝냄시간은 주어지지 않았다. 2점 경계값 문제 예제에서 경계조건을 다루는 방법을 적절히 고치면 이 방법은 여러점 경계값 문제에 쉽게 쓰일 수 있을 것이다.

Davidenko법은 시간 같은 독립변수가 연속 파라미터인 연속법의 하나로 보는 것이 가장 적절하고 결과 방정식은 Davidenko법을 특정의 대수 문제에 응용한 결과와 같다. 대수 문제가 특이 자코비안을 가지는 최적제어의 응용의 경우 답은 외현적 계 풀이 프로그램보다는 내재적 계 풀이 프로그램을 써서 찾아야 한다.

수렴영역은 아주 작은 문제에서도 정할 수 없었으나 Edelen의 연구는 결정방법에 대한 희망적인 생각을 가지게 한다. 상호상대방정식이 만들어지는 과정은 이를 뒷받침한다. 비연속성의 효과에 대한 조사는 예제에서 다른 정도만 이루어졌으나 적어도 사용된 성긴 격자에서는 signum 함수를 해석적인 비슷한 함수로 바꾸는 것은 필요하지 않았고 입력스위칭을 특별히 다루지 않고도 답을 찾을 수 있었다. 경계조건을 포함시키려면 더 많은 대수 조작을 해야 하나 예제에서 보듯 더욱 강인해짐을 알 수 있다.

Pontryagin의 최대원리에서 끌어낸 제어법칙은 스위칭시간을 정확하게 찾아야 하는 것이 그 특징인데 저자의 특정 목적은 적절한 스위칭구간을 아는 것으로 충분했다. 같은 기법의 다른 사용자들은 스위칭시간의 계산을 좋게 하려면 격자를 잘게 자르거나 보간법을 만들어야 할 필요가 있을 수도 있다.

Davidenko법은 이미 있는 표준 프로그램을 써서 단 순히 또다른 동적 시뮬레이션을 하는 것이기 때문에 구현하기가 쉽다.

참고문헌

1. Davidenko, D. F., "On a New Method for Numerical Solution of Systems of Nonlin-

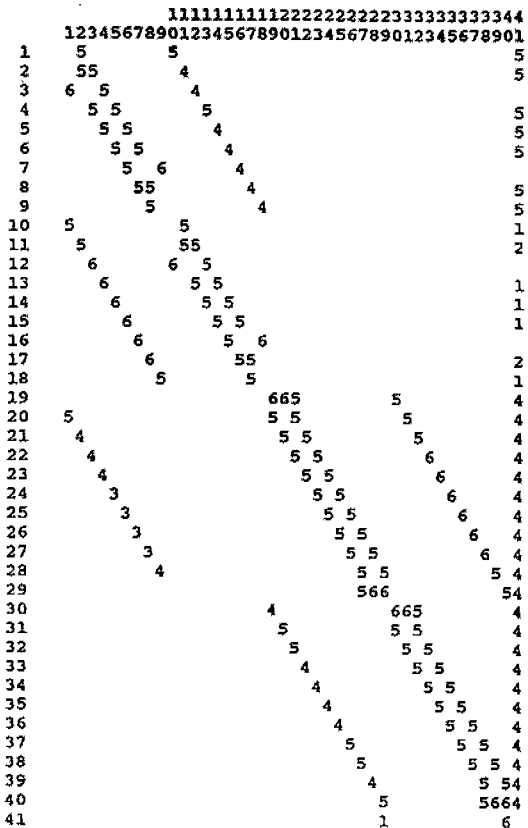


Fig. 5 Jacobian Matrix of Two-Point BVP of a Driven Pendulum Example

- ear Equations (Russian)", Doklady Akad. Nauk. SSSR, Vol.88, pp.601-602, 1953 (Journal translated as Soviet Math. Dokl.).
2. Edelen, D. G. B., "On the Construction of Differential Systems for the Solution of Nonlinear Algebraic and Transcendental Systems of Equations", Numerical Methods for Differential Systems, Lapidus, L., and Schiesser, W. E. (eds), Academic Press, New York, pp.67-84, 1976.
 3. Schiesser, W. E., "Application of Davidenko's Method in Chemical Engineering", presented at the 2nd World Congress of Chemical Engineers, October 1981.
 4. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., The Mathematical Theory of Optimal Processes, The Macmillan Press, New York, 1964.
 5. Paiewonsky, B., "Optimal Control: A Review of Theory and Practice", AIAA Journal, Vol. 3, No. 11, pp.1985-2006, 1965.
 6. Athans, M., "The Status of Optimal Control Theory and Applications for Deterministic Systems", IEEE Transactions on Automatic Control, Vol.AC-11, No.5, pp.580-596, June 1956.
 7. Kahn, M. E., The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains, PhD dissertation, Stanford University, December 1969.
 8. Weinreb, A. and Bryson, A. E., "Minimum-Time Control of a Two-Link Robot Arm", IFAC Control Applications of Nonlinear Programming and Optimization, Capri, Italy, pp.195-199, 1985.
 9. Geering, H. P., Guzzella, L., Hepner, S. A. R., and Onder, C. H. "Time Optimal Motions of Robots in Assembly Tasks", IEEE Transactions on Automatic Control, Vol.AC-31, No.6, pp.512-518 June 1986.
 10. Nakamura, Y., and Hanafusa, H., "Optimal Redundancy Control of Robot Manipulator", The International Journal of Robotics Research, Vol.6, No.1, pp.32-42, Spring 1987.
 11. Ojika, T., and Kasue, Y., "Initial-Value Adjusting Method for the Solution of Nonlinear Multipoint Boundary-value Problems", Journal of Mathematical Analysis and Applications, Vol.69, pp.359-371, 1979.
 12. Talisa, S. H., "Application of Davidenko's Method to the Solution of Dispersion Relations in Lossy Waveguiding Systems", IEEE Transactions on Microwave Theory and Techniques, Vol.MTT-33, No.10, pp.967-971, October 1985.
 13. Schiesser, W. E., DSS/2:Differential System Simulator, Version 2, Lehigh University, 1976.
 14. Hindmarsh, A. C., "LOSDE and LSODI, Two New Initial Value Ordinary Differential Equation Solvers", ACM SIGNUM Newsletter, Vol.15, No.4, pp.10-11, December 1980.
 15. International Mathematical and Statistical Libraries, IMSL Library, IMSL, Inc., Houston, TX, 1980.
 16. Petzold, L. R., "A Description of DASSL: A Differential/ Algebraic System Solver", Scientific Computing, Stepleman, R., et al. (eds), North-Holland, pp.65-68, 1983.
 17. Byrne, G. D., and Hindmarsh, A. C., "Experiments in Numerical Methods for a Problem in Combustion Modeling", Applied Numerical Mathematics, Transactions of IMACS, Vol.1, No.1, pp.29-57, January 1985.
 18. Hindmarsh, A. C., and Johnson, S. H., "Dynamic Simulation of Reversible Solid-

- Fluid Reacting in Nonisothermal Porous Spheres with Stefan-Maxwell Diffusion", Chemical Engineering Science, 1988.
19. Johnson, S. H., Harlow, D. G., and Yoon, J. S., "Time-Optimal Multistage Controllers for Nonlinear Continuous Processes", ASME Journal of Dynamic Systems, Measurement, and Control, Vol.108, No.3, pp.240-247, 1986.
20. Dissinger, G. R., Schiesser, W. E., and Johnson, S. H., "Large Scale Dynamic Process Simulation Experience: Effect of Jacobian Structure on Integrator Performance", Advances in Computer Technology, ASME Century II Publication, Vol.1, pp.50-60, 1980.