목적 함수의 연립 방정식화를 위한 직접 도함수 산출에 의한 최적치 계산법

김 주 홍*, 엄 기 환*

Optimal Algorithm from Object Function to Simultaneous Equations by Direct Derivative

Joo-Hong Kim, Ki-Hwan Eom

요 약

최적 제어나 최적 설계에 사용되는 목적함수를 연립 방정식화하여, Newton법에 의하여 최적치를 구하는 알고리즘을 제안하였다. 제안한 방식은 도함수의 산출과 입력이 불필요한 일반 도함수를 프로그램화한 직접 미분법(DDA)에 의하여 목적함수와 초기치만을 입력하여 최적치를 구하는 간단한 방식이다. 제안한 방식을 최적 제어와 최적 설계에 적용하여 유용성을 확인하였다.

Abstract

In this paper, we propose an algorithm for the object function used in optimal control or optimal design that sets the object function as simultaneous and then calculates the optimal value according to the Newton method. The proposed method calculates the optimal value simply using two inputs; object function and initial value, using the DDA(Direct Derivative Algorithm) which is a programmed ordinary derivative function that does not inquire the calculation of derivative function nor any inputs. And we have verified the usefulness of the algorithm for the optimal control and optimal desing.

* 동국대학교 전자공학과 교수
접수일자 : 1999년 10월 29일

1. 서 론

최근에 시스템이나 장치의 설계 및 제어 등의 최적화 문제는 이공학 분야뿐만 아니라 사회 과학 분야 등 여러 분야에서 필요성이 점점 더 증대되어 가고 있다. 최적화 문제는 독립된 매개 변수를 설정하고, 이들 매개 변수를 가지고 문제에 적절한 목적함수를 만들어 목적 함수가 극치(최소치 또는 최대치)가 되는 매개 변수의 값을 산출하는 것이다. 일반적으로 목적 함수는 비선형 다변수로 구성된 1개의 범함수로서, 이 함수의 극치(최소치 또는 최대치)를 산출하는 것은 쉽지 않으며, 이에 대한 알고리즘은 간접 탐색법과 직접 탐색법으로 대분된다.[2][3][6].

간접 탐색법은 목적 함수의 기울기와 이동거리, 탐색 방향 및 보조 상태 방정식 등의 계산을 수행하여 탐색점을 결정하므로, 한 개의 탐색점을 결정하기 위한 계산량이 많고, 프로그램이 매우 복잡하며, 결과치의 기복이 있어 때로는 발산하는 등의 문제점이 있다. 그러나 목적함수를 계산하는 회수가 비교적 적고, 계산 시간이 짧은 장점이 있다. 이 탐색법은 CGD(Conjugate Gradient Descent), 최급강하법(Steepest Descent) 등이 있으며 실시간 제어 등에 사용되고 있다.[5][8].

직접 탐색법은 설정한 탐색 구간 내에서 탐색점을 결정하고, 탐색점에서 직접 목적 함수를 계산하여 극치를 찾아내므로 단위 계산량은 적으나 반복 계산이 필요하므로 시간이 많이 소요되는 문제점이 있다. 그러나 프로그램이 간단하며, 결과치의 기복이 발생하지 않으므로 발산이 되지 않는 장점이 있다. 이 탐색법에는 Golden section 탐색법, 좌표 변환법 등이 있다[2][5].

이상과 같이 탐색법들에는 많은 반복 계산을 필요로 하거나 또는 유한 차분 구배값들의 계산에서 발생하는 버림 또는 반올림 오차와 같은 수치적인 문제에 민감하게 거동할 수 있으므로 컴퓨터에 입력하는 프로그램의 작성에 많은 노력이 필요하다[1][4][7].

본 논문에서는 1개의 함수인 목적 함수를 이용하여 다변수 연립 방정식을 도출하고, 도출한 연립 방정식을 Jacobian을 사용한 벡터형 Newton법을 사용하여 최적치를 산출하는 매우 간단하고 편리하며 정확한 방식을 제안한다. 제안한 방식은 직접 도함수(Direct Derivative) 계산법을 사용하므로 목적 함수와 초기치만을 입력시키고, 개별 문제에 따른 식의 유도와 프로그램화 등의 일체 절차 없이 해결되는 새로운 산출 방식이다. 제안한 알고리즘의 실용성을 확인하기 위하여 직접 도함수법의 성능, 최적치 탐색이 어려운 시험함수로 Rao 및 Rosenbrock의 banna 함수 등에 적용한다[2][5][6]. 또한 DC 모터의 최적 위치 제어기 설계 및 MOSFET 전력 증폭기의 최적 설계 문제에 적용하여 다른 탐색법들과 비교 검토하여 극히 간편하고, 정확성이 큰 산출 방식임을 확인한다.

2. 직접 도함수법에 의한 연립 방정식화

목적 함수는 최적 제어나 최적 설계들의 분야에서 계통, 목표, 제약 조건에 따라 구성이 다르며, 이것의 최적치를 계산하는 독립변수 즉 매개 변수로 구성된 함수로서 이것의 극치를 만드는 매개 변수의 수치 즉 최적치를 구하는 것이 최적 수치 계산이다[5].

일반적으로 목적 함수는 매개 변수의 범함수로 구성된 비선형 다변수 함수로서, 극소치를 취한 경우가 많으며, 이때는 부의 수가 없도록 주로 자승항으로 구성한다. 이의 최적치 수치 계산은 매우 복잡하며 많은 계산법이 있으나 일반적으로 산출법이 간결하지 않다[5].

여기서는 목적함수로부터 연립 방정식을 작성하고, 이 연립 방정식의 해가 매개 변수의 최적치인 것을 이용한 것으로, 개별 문제에 대한 산출 방식에 미분계수나 보조 계산 등을 도출하여 입력하는 복잡성이 없으며, 직접 도함수 계산법을 이용하여, 일반적 산출 알고리즘을 작성하여서 목적 함수와 초기치만의 입력으로 해를 얻는 간단한 산출 방식이다.

2-1 직접 도함수 알고리즘(Direct Derivative Algorithm : DDA)

변수 x의 함수 $y=f(x)$의 도함수 $y' = df(x)/dx$

의 정의는 식(2-1)과 같으며,

$$y' = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \qquad \cdots\cdots (2\text{-}1)$$

여기서 h는 x의 편차이다. 그러나 식(2-1)에 관한 계산은 계산기의 성능에 따라 $h\to 0$이 되지 못하고 그 크기에 제한이 되므로 식(2-2)와 같이 쓸 수 있으며, 식(2-1)과는 오차가 있다.

$$y' \fallingdotseq \frac{f(x+h)-f(x)}{h} \qquad \cdots\cdots (2\text{-}2)$$

또한 식(2-2)는 x점에서의 편차 h를 1/2로 택한 방식으로 표시하면, 식(2-3)이 된다.

$$y' \fallingdotseq \frac{f(x+\frac{h}{2})-f(x-\frac{h}{2})}{h} \qquad \cdots\cdots (2\text{-}3)$$

이때 정확성은 식(2-2)에 비하여 식(2-3)이 크다.

식(2-3)을 이용하여 원함수 $f(x)$와 x의 초기치만 입력시키면 직접 도함수가 산출된다. 일반적으로 n계 도함수 y^n는 n회 도함수의 도출 절차를 거쳐서 작성된 도함수를 입력시켜서 계산하고 있으나, 여기서는 지정된 함수가 불필요한 일반적인 고계 도함수의 알고리즘을 다음과 같이 작성하여서 직접 계산을 가능하게 한다.

n계 도함수 y^n을 나타내면 식(2-4)와 같으며,

$$y^n \fallingdotseq \frac{1}{h^n}\sum_{i=0}^{n}(-1)^i\,{}_nC_i f[x+(n-i)h] \qquad \cdots\cdots (2\text{-}4)$$

주어진 함수 $y=f(x)$만 가지고 n계 도함수의 직접 산출이 가능하다. 그러나 정확성이 보다 우수한 식(2-3)을 이용하기 위하여 식(2-4)를 다시 정리하면 식(2-5)와 같다.

$$y^n \fallingdotseq \frac{1}{h^n}\sum_{i=0}^{n}(-1)^i\,{}_nC_i f[x+(\frac{n}{2}-i)h] \qquad \cdots\cdots (2\text{-}5)$$

식(2-5)를 n계 도함수의 직접 도함수 알고리즘의 계산 공식으로 도출한다. 또한 x_k의 다변수를 갖는 함수에 대한 n계의 편미분 계수를 구하는 직접 미분 산출법의 일반식은 식(2-6)과 같다.

$$\frac{\partial y^n}{\partial x_k^n} \fallingdotseq \frac{1}{h^n}\sum_{i=0}^{n}(-1)^i\,{}_nC_i f[x_1,x_2,\cdots,x_k+(\frac{n}{2}-i)h,\cdots,x_n] \qquad \cdots\cdots (2\text{-}6)$$

따라서 식(2-6)을 이용하여 만든 프로그램을 사용하면 원함수와 초기치만으로 개별의 도함수 산출이 가능한 직접 도함수 알고리즘이 성립하여, Newton-Raphosn법이나 Jacobian에서 사용되는 미분 계수 그리고 편미분 계수를 일일이 구하여 입력시켜야 되는 불편성을 피할 수 있다.

2-2 연립 방정식화

연립 방정식화의 과정은 다음과 같다.

먼저 목적 함수 S를 식(2-7)이라고 하고

$$S(X)=f(x_1,x_2,x_3,\cdots,x_n) \qquad \cdots\cdots (2\text{-}7)$$

여기서 X는 $x_1,x_2,\cdots,x_n$의 독립 변수(매개 변수)를 성분으로 한 벡터이며, 이들 각각은 또 다른 독립 변수의 함수로 범함수의 경우가 많다.

기존의 산출 방식으로 다변수 x의 함수 $y=f(x)$에 관한 극치 계산법은 식(2-8)과 같은 방정식의 해인 x^*가 최적변수로 된다.

$$y'=\frac{df(x)}{dx}=0 \qquad \cdots\cdots (2\text{-}8)$$

이때 2계 도함수 y''가 $y''<0$인 경우는 y가 최대치, $y''>0$인 경우는 y가 최소치를 갖게 되며, $y''=0$인 경우는 만곡선이 되어서 극치가 되지 못하므로 식(2-8)에서 구한 x^*는 최적치가 되지 못한다. 따라서 식(2-7)로 주어진 목적 함수 S의 최적변수 $x_1^*,x_2^*,\cdots,x_n^*$의 계산법은 함수 S의 극치 부근에서 S의 미분 dS가 0이 되어야 하므로, 식(2-9)의 계수가 모두 0이 되어야 한다.

$$\frac{dS}{dX}=\frac{\partial f}{\partial x_1}+\frac{\partial f}{\partial x_2}+\cdots+\frac{\partial f}{\partial x_n}=0 \qquad \cdots\cdots (2\text{-}9)$$

즉 식(2-10)이 성립되어야 한다.

$$\frac{\partial f}{\partial x_1}=0,\ \frac{\partial f}{\partial x_2}=0,\ \cdots,\ \frac{\partial f}{\partial x_n}=0 \qquad \cdots\cdots (2\text{-}10)$$

그러므로, 식(2-10)에서 연립 방정식이 구성되고, 이의 해가 목적함수 S의 최적치가 된다.

여기서 각각의 편미분을 식(2-11)이라 하면,

$$y_1 = \frac{\partial f}{\partial x_1},\ y_2 = \frac{\partial f}{\partial x_2},\ \cdots,\ y_n = \frac{\partial f}{\partial x_n} \quad \cdots (2\text{-}11)$$

$\frac{\partial y_i}{\partial x_i} > 0$ 일 때 S가 최소치이고, $\frac{\partial y_i}{\partial x_i} < 0$ 일 때 S가 최대치로 되며 기타의 경우는 S의 극치가 되지 못한다. 단 i는 1, 2, …, n이다.

기존의 방식은 주어진 개별의 목적 함수와 편미분 계수를 유도하여 프로그램에 입력시켜서 식(2-9)를 산출하는 방법이다.

여기서는 식(2-12)와 같은

$$y_i'(x_1, x_2, \cdots, x_n) = \frac{f(x_1, x_2, \cdots, x_n + \frac{\Delta x_i}{2}, x_j, \cdots, x_n)}{\Delta x_i} - \frac{f(x_1, x_2, \cdots, x_n - \frac{\Delta x_i}{2}, x_j, \cdots, x_n)}{\Delta x_i} \quad \cdots\cdots (2\text{-}12)$$

일반식을 프로그램시켜 $f(x_1, x_2, \cdots, x_n)$의 함수를 입력하면 컴퓨터에 의해 직접 편미분 계수를 산출시키는 방법에 의한 계산법으로 목적함수 $f(x_1, x_2, \cdots, x_n)$만 입력하면 식(2-10)이 산출되는 직접 도함수법을 이용한다.

3. 직접 도함수법에 의한 연립 방정식의 해법

비선형 함수로 구성된 다변수 연립 방정식의 해법으로 Newton법이 있으며, 이 방법은 Jacobian J를 사용한 벡터 행렬로 구성된 식(3-1)의 반복 계산법으로, k는 계산 반복 회수이다[1][2].

$$X_k = X_{k-1} - [J_k(x)]^{-1} Y_k(X) \quad \cdots\cdots (3\text{-}1)$$

여기서 Jacobian J는 식(3-2)이며,

$$J = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \frac{\partial y_n}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_n} \end{pmatrix} \quad \cdots\cdots (3\text{-}2)$$

n개의 변수에 대하여서는 n^2개의 편미분 계수로 구성되므로, 이것을 도출하고 입력시키는 종래의 방법은 대단히 복잡하다.

제안한 방식은 Jacobian을 작성하여 입력시키지 않고 직접 프로그램하여 컴퓨터 자신이 처리하는 직접 미분법이다. 이것으로 범산식을 작성하여서 사용하면 Jacobian은 물론 벡터 행렬형의 Newton 방법에 관한 일반형 알고리즘에 의하여 개별함수와 변수만의 입력으로 해답이 얻어진다.

제안한 직접 도함수법에 의한 Jacobian의 일반적인 산출법은 다음과 같은 방법으로 프로그래밍한다.

전항에서 구한 목적 함수 $S(X)$의 연립 방정식인 식(2-10)을 이용하여 $y_i(x)$의 직접 편미분 계수 $\frac{\partial y_i}{\partial x_i}$를 구하는 방식을 정의하면 식(3-3)과 같다.

$$y_i' = \frac{\partial y}{\partial x_i} = \frac{y_{i+} - y_{i-}}{h} \quad \cdots\cdots (3\text{-}3)$$

단 h는 x의 구분 없이 공통되는 x의 미분치 Δx이다 Jacobian의 인수 $\frac{\partial y_i}{\partial x_j}$도 같은 방식을 써서

$y_{ij} = \frac{\partial y_i}{\partial x_j} = \frac{\partial^2 f_i}{\partial x_j \partial x_i}$ 는 식(3-4)에서

$$\begin{aligned} y_{ij+} &= y_i(x_1, x_2, \cdots, x_j + \tfrac{1}{2}h, x_{j+1}, \cdots, x_n) \\ y_{ij-} &= y_i(x_1, x_2, \cdots, x_j - \tfrac{1}{2}h, x_{j+1}, \cdots, x_n) \end{aligned} \quad \cdots\cdots (3\text{-}4)$$

식(3-5)와 같으므로

$$\frac{\partial y_i}{\partial x_j} = \frac{y_{i+} - y_{i-}}{h} \quad \cdots\cdots (3\text{-}5)$$

직접 프로그램에 의해 일반적인 방식으로 산출되고, 식(3-1)에 의하여 최적치가 산출된다. 즉 목적 함수 S(X)와 매개변수 X만의 입력으로 해답이 얻어지는 알고리즘이 된다. 이것에 관한 플로우차트는 그림 1과 같다.

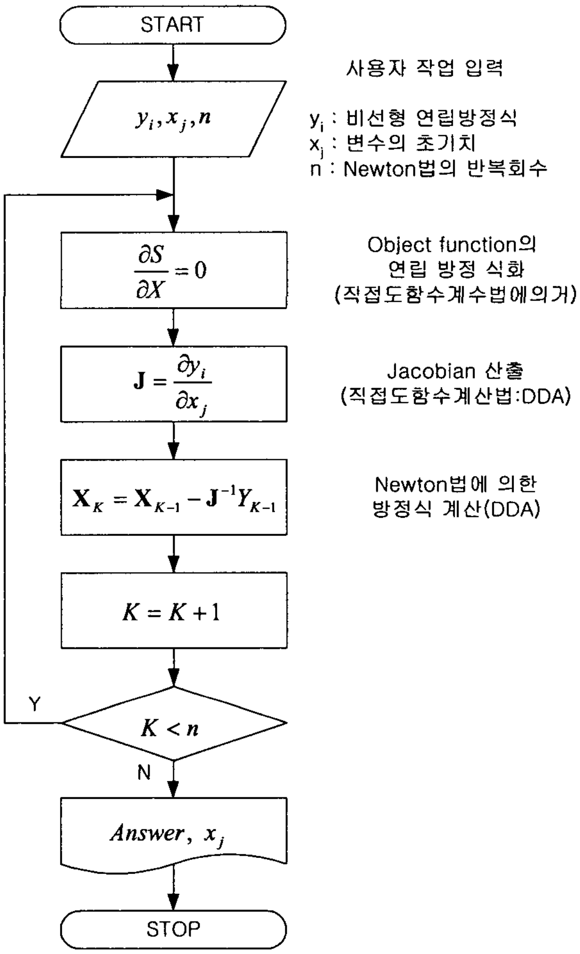

그림 1. 프로우 차트

Fig. 1. Flow Chart

4. 응용 예

제안한 알고리즘의 신뢰성을 확인하기 위하여 C++로 프로그램을 작성하여 먼저 직접 도함수법의 성능과 각종 시험 함수들을 목적 함수로 적용한 경우 및 최적 제어와 최적 설계법에 대한 예를 PC 586으로 시뮬레이션한다.

4-1 직접 미분법의 성능

임의의 함수에 대한 고계도함수의 미계수를 구하기 위하여 제안한 직접 미분산출 공식인 식(2-5)를 이용하여 계산한다. 한 개의 변수로 이루어진 n계 도함수의 미계수를 직접 산출하기 위하여 사용한 함수는 $y=\sin x$, $y=x^3$ 그리고 $y=e^x$ 이며, 각각 개별의 변수에 대한 값은 $x=\frac{\pi}{4}$, $x=2$ 그리고 $x=1$로 취하고, 5계 도함수까지 구하여 계산한 값은 표4.1과 같다.

표 4.1. 직접 도함수법에 의한 함수의 고계도함수 계산치

Table 4.1. The higher order computation value by the direct derivative method

함수	n계 도함수	식(2-5) 사용		식(2-4) 사용	
		계산치	오차	계산치	오차
$y=\sin x$ 단 $y=\frac{\pi}{4}$	1	0.707107	0.000000	0.707107	0.000000
	2	-0.707106	0.000001	-0.707114	0.000007
	3	-0.707460	-0.000353	-0.706045	0.001062
	4	0.707095	-0.000012	0.721095	0.013988
	5	0.710618	-0.003511	0.689193	-0.017913
$y=x^3$ 단 $x=2$	1	12.000001	0.000000	12.000001	0.000000
	2	12.000008	0.000007	12.000054	0.000053
	3	6.000001	-0.000001	6.000004	-0.000004
	4	0.000001	-0.000001	0.000002	-0.000002
	5	0.000174	-0.000174	0.000444	-0.000444
$y=e^x$ 단 $x=1$	1	7.718282	0.000000	2.718282	0.000000
	2	2.718284	0.000004	2.718303	0.000021
	3	2.716923	-0.0013858	2.722362	0.004080
	4	2.718327	0.000046	2.773241	0.054959
	5	2.704762	-0.013519	2.787160	0.068878

표4.1 에서와 같이 제안한 직접 미분 산출 공식인 식(2-5)를 사용한 경우가 식(2-4)를 사용한 경우보다 정확성이 크며 이론치에 접근한다. 또한 도함수의 계수가 증가되면 오차가 커지나, 사용한 함수의 3계 도함수까지 오차는 이론수치의 수만 분의 일 이하이므로 정확성을 갖고 있다.

4.2 목적함수로 사용한 시험함수

비선형 다변수인 일계 도함수로 구성된 함수는 목적함수와 같은 함수이므로 수학계에 널리 알려진 몇 개의 함수를 시험함수로 택하여 제안한 방식을 이용하여 계산한다.

첫째 Rao의 시험함수로 식(4-1)에 대한 최소치를

구하여 비교 검토한다[6].

$$J(X) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \quad \text{(4-1)}$$

표 4.2는 Rao의 시험함수를 제안한 방식과 CGD, Steepest descent, Golden section방식등과의 결과들과 비교한 것이다.

표 4.2 Rao의 시험함수의 최소치 계산 결과
Table 4.2 The results of mimimum values for the Rao's test function

방식 (알고리즘)	제안한 방식	CGD	Steepest descent	Golden section
최소치	x_1=-1 x_2=1.5	x_1=-1 x_2=1.5	x_1=-1 x_2=1.5	x_1=-1 x_2=1.5
목적함수의 최소치	-1.25	-1.2527	-1.525	-1.2581
계산회수	25	25	23	36
계산시간	0.31sec	0.31sec	0.28sec	0.36sec

두 번째는 식(4-2)와 같은 Rosenbrock의 banana 함수를 시험함수로 하였다[8].

$$J(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \quad \text{(4-2)}$$

표 4.3은 banana함수의 최소치 계산 결과이다.

표 4.3 banana 함수의 최소치 계산결과
Table 4.3 The results of minimum values for a banana function

방식 (알고리즘)	제안한 방식	CGD	Steepest Descent	Golden Section
최소치	x_1=1.0 x_2=1.0	x_1=1.0 x_2=1.01	x_1=1.0 x_2=1.0	x_1=1.01 x_2=1.01
목적함수의 최소치	1e-18	1.525e-13	1.5545e-13	1.25e-10
계산회수	53	60	63	238
계산시간 (sec)	0.58sec	1.00sec	1.02sec	3.07sec

이상과 같이 시험함수의 예를 보면 제안한 적접도함수법에 의해 계산된 목적함수의 최소치가 더 최소이며 최적치도 유사하므로 유용성을 학인할 수 있다.

4-3 최적 제어 예 (DC 모터 위치 제어)

DC 모터를 구동장치로 하여 추적이 가능한 최적 위치 제어의 예를 시뮬레이션 한다.

DC 모터의 전달함수를

$$G(s) = \frac{34.2}{s(s+2.23)(s+5)} \quad \text{(4-3)}$$

로 한다[5][9][10]. 이를 이용하여 순방향 보상기를 연결한 위치 제어 시스템의 블록 선도는 그림 2와 같다.

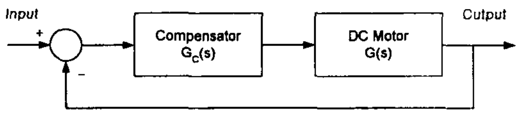

그림 2. 위치 제어 시스템 블록 선도
Fig. 2. Position Control System Diagram

그림 2의 위치 제어 시스템의 특성을 향상시키기 위하여 식(4-4)의 전달함수를 갖는 진상 보상기를 연결한다.

$$G_c(s) = \alpha \cdot (s + \beta) \quad \text{(4-4)}$$

식(4-3)과 식(4-4)를 결합한 보상기 위치 제어 시스템을 설계하면 식(4-5)의 전달 함수를 갖는다.

$$G_c(s)G(s) = \frac{34.2 \cdot \alpha \cdot (s+\beta)}{s(s+2.23)(s+5)} \quad \text{(4-5)}$$

식(4-5)를 상태 방정식으로 표현하면 식(4-6)과 같다.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 34.2 \cdot \alpha \cdot (\gamma - x_1 + x_3) - \beta \cdot x_2 \\ \dot{x}_3 &= (\beta - 5)(\gamma - x_1) - 5x_3 \end{aligned} \quad \text{(4-6)}$$

여기서 x_1은 위치, x_2는 속도, x_3는 가속도를 나타내는 변수이다. 이때의 목적 함수는 식(4-7)이다.

$$S = x_4(t_f) + w_1(\gamma - x_1(t_f))^2 + w_2(x_2(t_f)) \quad \text{(4-7)}$$

여기서 x_4는 적분오차를 나타내는 변수로서

$x_4=(r-x_4)^2$이고, t_f는 최종시간이며, w_1과 w_2는 하중상수로서 1000으로 한다.

목적함수인 식(4-7)은 매개변수 α, β의 최적치 계산용으로 시간함수인 $x_1, \cdots, x_4$가 있고, $x_1, \cdots, x_4$의 계산이 내포된 것으로 $S(\alpha, \beta, X(t))$의 복잡한 비선형 다변수 함수이다. 그러나 직접 도함수법으로 직접 계산이 가능하다.

제안한 알고리즘의 결과와 CGD(Conjugate Gradient Descent)법, 최급 강하법(Steepest Descent) 및 Golden Section 방식의 결과를 비교한 것은 표 4.4이다.

표 4.4 목적함수 최소치 계산 비교

Table 4.4 The Comparison of Object Function Minimum Value

방식 (알고리즘)	제안한 방식	CGD	Steepest Descent	Golden Section
최소치	α=0.4407 β=2.1621	α=0.4407 β=2.1621	α=0.4407 β=2.1621	α=0.4407 β=2.1621
목적함수의 최소치	0.2664	0.2940	0.28985	0.2995
계산회수	36	40	38	45
계산시간 (sec)	0.51	0.55	0.53	1.02

제안한 알고리즘의 목적함수의 최소치가 가장 적은 것을 알 수 있으며, α, β의 최소치도 서로 유사함을 확인할 수 있다.

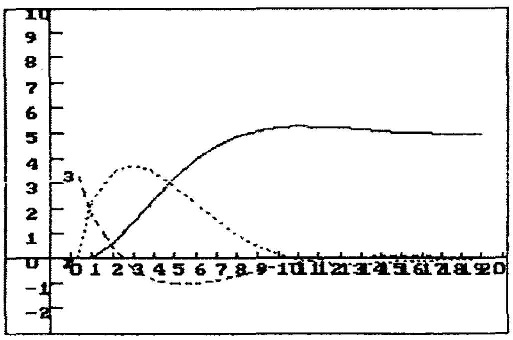

그림 3. 위치 제어 시뮬레이션

Fig. 3. Position Control Simulation

제안한 알고리즘으로 계산된 결과를 DC모터의 위치 제어에 시뮬레이션 한 것은 그림 3과 같다.

그림에서 수평축은 시간×0.2[sec], 수직축은 거리×0.2[ms]를 표시하고 있으며, 직선은 이동 거리[m], 작은 파선은 속도[m/s], 큰 파선은 가속도[m/s²] 응답 곡선이다.

4-4. 최적 설계 예 (MOSFET 전력 증폭기의 최적 설계)

각종 통신회로, 오디오 출력회로 등에 사용되는 MOSFET 전력 증폭기의 최적 설계를 한다. 특히 통신 분야에서는 전력 증폭기의 비선형 성분으로 인하여 출력에서 원치 않는 IMD(Inter Modulation Distortion), XMD(CrossModulation Distotion) 등의 왜율이 있으며, 이는 다중 채널 전송시 장해의 원인이 되어 시스템의 성능을 저하시키므로 이를 개선하기 위하여 선형화하고 있다.

왜율과 전력 손실에 대한 목적함수를 식(4-8)과 같이 설정한다.[11]

$$
\begin{aligned}
J &= W_1 E_{dis}(V_a, V_{gm}) + W_2 P_{loss}(V_a, V_{gm}) \\
&= W_1 2R_0K^2\left[\frac{(V_a+V_{gs})^5}{5} - \frac{32}{5}V_a + 4V_a^3V_{gs}^2 - 2V_aV_{gs}^4 + 6V_a^5\right] \\
&\quad + W_2\left[\frac{2}{\pi}V_{dd}K^2(V_a+V_{gm})^3 - R_0\left\{K^2(V_{gm}+V_a)^5 - \frac{32V_a^5}{5}\right\} - 2KV_a^3\right] \qquad \cdots\cdots (4\text{-}7)
\end{aligned}
$$

여기서 W_1은 왜율의 하중상수, W_2는 전력손실의 하중상수, V_a는 바이어스 전압, V_{gm}은 게이트 소스 최대전압, R_0는 부하저항, K는 MOSFET의 특성에 관계되는 상수이다.

표 4.5. 목적함수 최소치 계산 비교

Table 4.5. The Comparison of Object Function Minimum Value

방식 (알고리즘)	제안한 방식	CGD	Steepest descent	Golden section
최소치 Va[V]	3.88	3.86	3.88	3.87
목적함수의 최소치	315.031	398.118	316.25	333.6712
계산회수	72	72	69	78
계산시간[sec]	1.33	1.34	1.21	1.87

시뮬레이션 조건으로 부하에 최대 25[W]의 전력을 공급하는 증폭단을 설계할 때 최대 부하 전류 3[A]로 주어 최적 바이어스 전압을 구한다. 게이트 소스간 전압을 8[V]로 설정하여 하중 상수 $W_1 = W_2 = 1$로 놓고 목적함수 최소치 계산 결과를 비교한 것은 표 4.5이다.

표 4.5에서와 같이 제안한 방식은 최적 설계에서도 좋은 결과를 얻은 것을 확인하였다.

5. 결 론

목적함수의 최적치 계산에 대한 새로운 알고리즘을 제안하였다. 제안한 알고리즘은 목적함수로부터 다변수 연립 방정식을 계산하고, Jacobian을 사용한 벡터형 뉴턴법에 의하여 최적치를 구한다. 산출 방법은 일반형 편미분 계수 알고리즘을 이용하여, 개별함수에 따른 도함수 입력과정이 필요 없이 직접 도함수법에 의하여 원함수인 목적함수와 초기치 만의 입력으로 최적치가 산출되는 계산법으로 이의 특성은 다음과 같다.

1. 직접 도함수법은 미분계수를 필요로한 모든 수학의 수치계산에서 미분계수를 산출 입력하지 않고, 제시된 원함수와 초기치만 입력시켜서 고계 도함수를 컴퓨터 프로그램으로 직접 구할 수 있는 극히 간결하고 정확한 알고리즘이다.
2. 목적함수의 편도함수를 0으로 한 연립 방정식을 사용하여 뉴턴법으로 목적함수의 최적치를 구한다.
3. 목적함수의 극치 계산은, 비선형 다변수 연립 방적식의 일반적인 해법에서와 같이 편도함수를 구할 필요 없이, 목적함수와 변수의 초기치만을 입력하여서 직접 최적치를 산출하는 새로운 수치계산법으로 합리적이고, 간편하고, 정확성, 안정성 등에서 종래의 방식보다 우수하다.

이상과 같이 종래에 개발된 알고리짐에 비해 극히 간편하고 정확하며 실용성이 큰 새로운 알고리즘임을 확인하였다.

참고문헌

[1] SHANS KUO, Computer Applications of Numerical Methods, Addison Wesley, 1972

[2] LAWRENCE HASDORFF, Gradient Optimization and Nonlinear Control, John Wiley & Sons Inc., 1976

[3] Lee W. Johnson, R. Dean Riess, Numerical Analysis, Addion Wesley, 1982

[4] Curtis F. Gerald Partick O. Wheatly, Applied Numerical Analysis, Addison Wesley, 1983

[5] G. V. Reklaitis, A. Ravindrau and K. M. Ragsdell, Engineering Optimization Methods and Applications, John Wiley and Sons. New York, 1983

[6] S. S. Rao, "Optimization Theory and Applications." Halsted Press, 1984

[7] T. A. Synmau and L. P. Fatti, "A Multi-Start Global Minimization Algorithm with Dynamic Search Trajectories," JOTA, Vol. 54, No.1, pp.121-141, July 1987

[8] Jasbir S. Arora, Introduction to Optimum Design, McGraw-Hill Book Company, 1989

[9] S. A. Aisagaliev, "Optimal Control of Linear Systems with Fixed Trajectory Endpoints and Bounded Control," Differential Equations, Vol. 32, No.8, pp. 1017-1028, 1996

[10] David G. Hall, "Convention of Optimal Control Problems into Parameter Optimization Problems," Journal of Guidance, Control, and Dynamics, Vol. 20, No.1, Jan. 1997

[11] 김주홍, 엄기환, 이원일, 이용구, 손동설, "최적 바이어싱에 의한 MOSFET 선형 전력 증폭기의 설계," 한국통신학회지, 제20권, 제11호, pp.11-19, 1995

김 주 홍(Joo-Hong Kim)
1948년 6월 목포중학교 졸업 (6년제)
1952년 9월 서울대학교 공과대학 전자공학과 졸업(공학사)
1981년 8월 전북대학교 대학원 전기공학과 졸업 (공학박사)
1952년 3월~1954년 9월 국방부 과학기술연구소 전기과 연구원
1954년 10월~1965년 10월 조선대학교 전기공학과 전임강사
1963년 12월 제2회 전기주임기술자 1급 합격
1964년 12월 제1회 기술사 합격(공업계측제어)
1965년 11월~1986년 2월 한국 집진기 공업주식회사 창립
1968년 3월~1994년 2월 동국대학교 전자공학과 교수
1968년~1980년 대한전기주임기술자협회 이사
1978년~1979년 공군과학자문위원(공군참모총장)
1978년~1980년 신용보증기금 기술 및 경영지도위원
1982년 한국공업표준협회 품질관리상담위원(전자분야)
1982년~1991년 생산기술연구원 자문위원
1983년~1984년 일본 나고야대학 전기과 객원교수
1983년~1986년 일본 고주파열연연구소와 공동연구
1989년 8월~1992년 7월 한국코아 주식회사 연구소 고문
1976, 1982, 1990, 1993년 공업표준심의회 위원(공업진흥청장)
1993년 6월 22일 ~ 23일 기술개발시범기업 WORKSHOP 지도강사(정전기 방전형 집진기)

엄 기 환(Ki-Hwan Eom)
1972년 2월 동국대학교 전자공학과(공학사)
1986년 2월 동국대학교 전자공학과(공학박사)
1978년 3월~1994년 유한대학 전기공학과 교수
1994년 3월~1999년 현재 동국대학교 전자공학과 교수
* 관심분야 : 인공지능 및 자동화 시스템 설계