한국공작기계학회 2000년도 춘계학술대회 논문집 pp.667~672

자율 이동 로봇의 실시간 제어를 위한 가·감속 방법의 개발

이 수 종*(창원대), 정 원 지*(창원대), 김 효 수*(창원대), 엄 태 진*(창원대)

Development of Acceleration/Deceleration Method for Real-time Control of Autonomous Mobile Robots

Soo Jong Lee*(Changwon Univ.), Won Jee Chung*(Changwon Univ.),

Huy Soo Kim*(Changwon Univ.), Tae Jin Eum*(Changwon Univ.)

Abstract

This article presents a new acceleration/deceleration method for real-time control of autonomous mobile robots. In this method, a function which produces the table of acceleration/deceleration in real-time is proposed. This function, while satisfying the basic concept of mechanics, can choose both various ranges of velocity and distance ranges for the selected velocities. Moreover it can control motors with real time. This function is convenient to be realized by programming. and it is faster than other functions because it can be made by assembly language.

1. 서 론

임의의 물체를 빠른 속도로 제어하기 위해서는 각각의 구간에 맞는 가·감속의 속도 계획을 세우게 된다. 그러나 이러한 속도 계획은 위치 제어와 맞물려 있고, 각각의 영역에 할당되는 구간이 달라서, 보다 효율적이고 빠른, 그리고 역학적인 해석에 맞는 속도 제어에 어려움이 있다. 로봇에서 사용할 동력원을 선택할 때에는 필요한 토크와 관성 부하 등을 고려해서 선택되는데 이렇게 해서 선택한 동력원은 이후로는 모두 역학적 해석에 의한 프로그램에 의존하게 된다.

그러나, 대부분 이러한 속도 제어의 프로그램과 서보 기기의 설정에 있어서 역학을 전공하는 이에 의해서 이루어지기 보다 이것을 제어하는 전기와 전자를 담당한 이들에 의해서 결정됨으로서 역학적 해석이 배제되어 왔으며, 이 때문에 동력원을 극한까지 사용하는 데에는 무리가 있어 왔다.
예로, 일반적으로 사용하는 로봇의 속도 제어 함수로는 사다리꼴 함수를 채택하는데, 이것의 장점은 이것을 프로그램으로 구현하기가 쉽고, CPU의 연산 처리 속도의 부담을 줄일 수 있다는 것이다. 이러한 연산 처리 방식은 매우 많이 알려져 있으며, 대부분의 모터의 속도 제어에 그 근간을 이루고 있다. 그러나 우리가 좀더 빠른 고속 제어와 위치 제어를 하고 싶을 때, 그리고 스텝핑 모터의 속도가 탈조 영역에 가깝게 들어섰을 때의 고속 제어에서는 반드시 역학적인 개념에 바탕을 둔 속도 제어를 하여야 한다. 특히, 인공지능 로봇에게는 가속 구간과 감속 구간, 그리고 그 속도 한계의 선택에 있어서 로봇에 자율권을 부여해야 함으로 사다리꼴 함수와 같은 간단한 가속 Table 생성 함수를 가지고는 이러한 요건을 만족시키는 데에는 그 한계가 있다.

본 논문에서는 인공 지능 로봇의 다양한 동적 움직임에 필요한 요건을 만족하고, 그리고 모터의 고속 제어를 위해서 필요한 역학적 개념과 이것을 연산 처리하는 제어기의 연산 속도의 부담을 줄이

기 위해서 사용할 수 있는 새로운 함수를 소개하고, 이 함수의 장점과 그 특징에 대해서 논하고자 한다.

2. 본 론

2.1 가·감속을 위한 조건

Robot이 가·감속을 하는 이유는, 임의의 주어진 일을 신속하게 수행하기 위하여 급격한 속도의 변화에서 일어나는 관성력을 제어하기 위해서 이며, 모터의 탈조 영역을 피하기 위해서이고, 각각의 Robot팔의 관절 부하를 최소한으로 하여 진동에 대한 소음을 줄이는 데 있다. 가·감속 제어 방식은 비단 Robot 분야에서 뿐만이 아니라, 신속한 운동을 요하는 거의 모든 기계 운동 메카니즘에 필수적이라고 하여도 과언이 아닐 것이다.

예로, 우리가 흔히 볼 수 있는 자동문 같은 경우에 문이 자체의 무게를 이기지 못하여 늦게 열린다거나, 혹은 빨리 열리게 하기 위해서 모터에 급격한 속도를 요하면 관성력 때문에 모터가 탈조를 일으키든지 혹은 자동문 각각의 요소에 무리를 일으켜서 제품의 수명을 짧아지게 하는 일이 생긴다. 그렇다면 "어떻게 자동문을 제어해야 높은 속도와 제품의 수명 연장, 이 두 가지 조건을 만족시킬 것인가?"하는 문제에 당면하게 된다. 이러한 문제는 비단 자동문뿐만이 아니라, 고속 전철에서 승객이 가속도에 저항을 민감하게 받지 않게 하면서 요구하는 속도에 가장 짧은 시간에 도달할 것인가, 혹은 Elevator에서의 '승객의 체감 중력과 속도 관계' 등과 같은 종류의 문제들과 그 맥을 같이한다.

이러한 문제는 속도 함수의 불연속 점을 제거함으로써 그 해답을 찾을 수 있다. Motor의 제어 그래프로서 사다리꼴 함수의 단점은 가속 구간에서 등속 구간으로 바뀔 때, 가속도가 불연속적으로 변함으로써 Robot의 Counter계산에 Error를 일으킬 수 있는 요소가 된다. 따라서 불연속 점을 제거하기 위해서는 우리는 가속도를 선형적으로 제어할 수 있는 방법을 모색하게 되는데,

첫 번째 조건으로 속도 함수를 미분한 가속도 함수가 불연속 점 없이 선형적으로 변하는 것이 필요하다.

두 번째로는 로봇의 속도를 계속해서 가속할 수 없으므로 등속을 요구하게 되는데 등속을 할 때의 가속도는 Zero가 됨으로 정지에서 가속이 시작되어 끝나는 양끝 구간은 반드시 가속도가 Zero가 되어야 한다. 예로 위의 Fig 2.1에서는 시작점과 끝점, 그리고 A점에서의 가속도가 Zero가 되어야 한다.

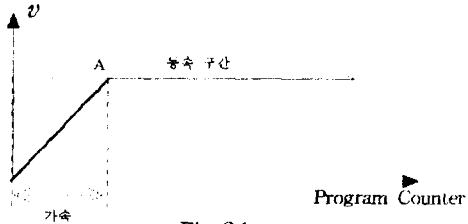

Fig 2.1

세 번째로는 이러한 함수는 CPU연산 처리 속도의 부담을 줄이고 실시간에 계속 변하는 가속 구간과 감속 구간에 맞는 속도 Table을 산출하기 위해서 Cpu의 연산에 가장 빠르게 적용된다고 할 수 있는 Assembly로 구현할 수 있어야 한다.

이러한 것을 만족한다면 Robot은 더욱 다양한 가속 구간과 속도 선택을 할 수 있어서 활성화된 움직임을 보일 수 있으며, 요즘 대두되고 있는 인공지능을 바탕으로 한 Robot의 동력학적인 움직임에 가장 알맞은 함수가 될 수 있을 것이다. 위의 내용을 간단하게 요약한다면, 속도 함수 $v(x)$는 전구간에서 1계 도함수 $a(x)$가 연속이어야 하며, 속도 함수 $v(x)$의 저크 $j(x)$가 전구간에서 유한 값을 가져야 한다.

2.2 기존에 알려져 있는 가·감속 함수

속도 함수의 선택에 있어서는 CAM설계를 하는 경우와 마찬가지로 Jerk 개념을 바탕으로 하게 되는데, 일반 함수의 선택에서는 4-5-6-7의 지수를 가지는 함수와, 초월 함수로는 단순 조화 함수와 그리고 사이클로이드 함수등을 사용한다. 그러나 이것은 이론적인 함수로서 우리가 역학적 요건에 맞는(가속도의 변화량이 연속적이게 하는) 함수를 선택했다고 하더라도 우리는 이러한 함수를 구현하기 위해서 Cpu가 연산 처리 속도에 대한 부담

을 고려해 보아야 한다. Cpu가 할 수 있는 연산 방법은 (연산 처리 속도가 빠른 Dsp일 지라도... ...)사칙 연산뿐이므로, 앞에서 언급한 함수들이 역학적 해석에는 맞을 지라도, 그것을 구현할 수 있는 연산 처리 속도에는 맞지 않는 부분이 발생하게 된다.

$$\sin x = x - \frac{x^3}{3!} + \cdots + \frac{(-1)^k}{(2k+1)!}(x)^{2k+1} + \cdots \qquad (2.2)$$

예로서 단순 조화 함수의 sine함수를 선택했다면, 이것이 컴파일 되어서 Rom에 입력될 때에는 사칙 연산에 맞는 멱급수 식 (2.2)와 같이 바뀌어서 입력되고 CPU가 이것을 표현하기 위해서는 수많은 곱셈 연산을 거쳐야 한다. 이러한 이유로 인해서 Robot 팔에 해당하는 각각의 조인트의 움직임은 조화 함수의 사용은 지양되고, 다항식 함수를 사용하는 것이 일반적이다. 그러나 이 다항식 함수에 사용되어 지는 곱셈 연산도, Cpu가 계산하는 다른 명령 수행에 비해 수십 배에 해당하는 시간 지연을 일으키기 때문에 실시간 제어를 하기에는 불가능하고, Robot이 움직이기 전에 궤적 계획을 세우고 입력한 후, 가속 영역과 등속 영역을 나누고, 그것에 맞는 속도 Table을 만든 후, 사용하는 방식을 택하게 된다.

3. 분수 함수의 소개

3.1 분수 함수의 계형 및 특징

연속 적인 미분에도 zero가 되지 않는 것은 비단, 조화 함수나 지수 함수가 아닐 지라도 일반 사칙 연산의 조합으로 표현할 수 있는 분수 함수가 있다. 본 논문에서 소개할 분수 함수의 형태는 식 (3.1-1)에서 볼 수 있듯이 총, 3회의 곱셈과, 1회의 나눗셈, 그리고, 1회의 덧셈으로 이루어져 있다. 이 함수를 속도 함수로 선택하였을 때의 역학적인 해석을 위해서 그래프의 요철과 그래프의 계형을 살펴 볼 필요가 있다.

$$v(x) = \frac{nx}{mx^2 + \alpha} \qquad (3.1\text{-}1)$$

$$v'(x) = \frac{\alpha n - mnx^2}{(mx^2 + \alpha)^2} = a(x) \qquad (3.1\text{-}2)$$

$$v''(x) = \frac{2mnx(mx^2 - 3\alpha)}{(mx^2 + \alpha)^3} = j(x) \qquad (3.1\text{-}3)$$

이 분수 함수는 식 (3.1-2) 에서 알 수 있듯 두 개의 극대 값을 $x = \pm\sqrt{\frac{\alpha}{m}}$ 에서 가지고, 세 개의 변곡점을 $x = 0, \pm\sqrt{\frac{3\alpha}{m}}$ 에서 가진다. 또한, 좌극한과 우극한이 Zero(0)으로 수렴하므로 이것을 바탕으로 그래프를 그리면 Fig.3.1과 같다.

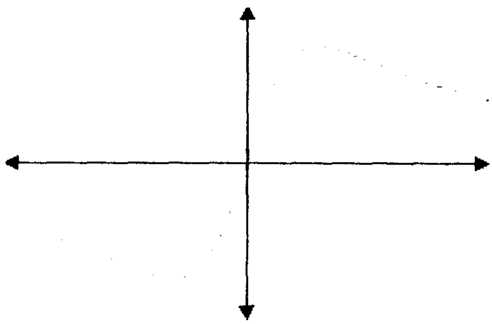

Fig 3.1

이 분수 함수의 그래프에서 우리가 사용할 수 있는 영역은 극대값과 극소값 사이의 영역의 구간만을 취하는데, 이러한 이유는 극점에서의 가속도가 zero(0)으로 진입하므로, 이 점을 기점으로 등속으로 진입할 수 있고, 다른 그래프와의 연결이 용이하기 때문이다. 따라서 그 결과적 그래프의 계형은 Fig3.2와 같다.(Fig3.2에서 수평좌표는 Counter의 축으로, 수직 좌표는 속도 축으로 나타내었다.)

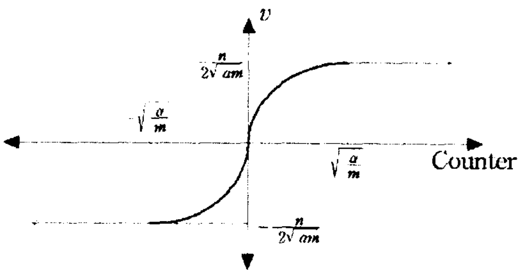

Fig 3.2

이 분수 함수를 속도 함수로 선택했을 때, 동

력위에 걸리는 힘을 계산하기 위해서, 가속도를 분석하면 $x=0$ 인 곳에가 가장 큰 힘을 받으며, 이때의 가속도는 $a(x)=v''_{x=0}=\frac{n}{a}$ 이다. 여기서 유의해야 할 것은 식(3.1-1) 에서의 속도 함수에서 상수 m은 가속도에 관여하지 않음을 알 수 있고, 변수 a가 커질수록 가속도는 작아 지고, n이 커질수록 가속도는 커진다. counter의 변화는 양수만을 가지므로 Fig 3.2의 그래프는 좌표 이동을 해야 한다. Fig 3.3과 식 (3.1-4)은 좌표 이동($\sqrt{\frac{a}{m}}, \frac{1}{2}\frac{n}{\sqrt{am}}$)을 한 최종적인 그래프와 수식이다.

$$v(x)=\frac{n(x-\sqrt{\frac{a}{m}})}{m\,(x-\sqrt{\frac{a}{m}})^2+a}+\frac{1}{2}\frac{n}{\sqrt{am}} \qquad (3.1\text{-}4)$$

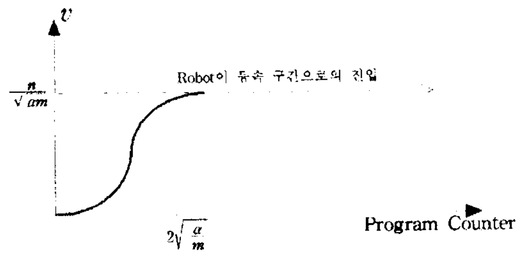

Fig 3.3

3.2 분수 함수의 V-A-J 선도

우리가 분수 함수를 속도 함수로 선택하였을 때 분수 함수의 역학적인 해석 및 적용을 위해서 가속도 선도와 Jerk선도를 그려본 후, 요철과 가속도의 연속과 불연속을 고찰해야 한다. Fig 3.4은 앞에서 설명한 분수 함수 식(3.1-4)의 V-A-J선도이다. Fig3.4 에서 볼 수 있듯이 이 분수 함수를 속도 함수로 선택하였을 때에는 속도와 가속도에서는 jump가 발생하지 않으나 가속에서 등속으로 진입하는 jerk선도에서 불연속 성을 보인다. 우리는 이러한 분수 함수의 결점을 보완하기 위해서 분수 함수 외에 또 다른 함수를 합성을 하여 jerk 선도의 연결점을 연속적으로 만들 필요가 있다.

3.3 분수 함수의 합성

속도 함수를 우리가 지향하는 형태를 만족시키기 위한 만들기 위해서는 수학 함수의 그래프 개형에 대한 어느 정도의 이해가 필요하다.

가속도($a(x)$)를 미분한 jerk 선도($j(x)$)가 연속적이기 위해서는 앞에서의 속도 함수($v(x)$)의 미분 함수 즉, 가속도 함수($a(x)$)의 극점과 변곡점이 일치되어야 하는 그래프 계형 상의 특징을 가지게 된다.

예로 Cam의 설계에 있어서 4-5-6-7 Polynomial을 선택하고, 1-2-3차 항이 모두 소멸되는 이유는 초기 조건을 만족시키기 위한 물리적 의미 외에도 수학적으로는 그래프의 계형 상 극점과 변곡점을 일치시켜서 3차 도함수(jerk 함수)의 연속성을 만족시키려는 중요한 이유도 그 중에 하나이다. 따라서 본 논문에서 사용하려는 분수 함수 또한 임의의 의 함수 $f(x)$를 더함으로써, 식(3.3)의 그래프 Fig 3.4에서의 가속도 함수($a(x)$)의 불연 속성을 막을 수 있어야 한다.

$$v(x)=\frac{nx}{mx^2+a}+f(x) \qquad (\text{식, } 3.3)$$

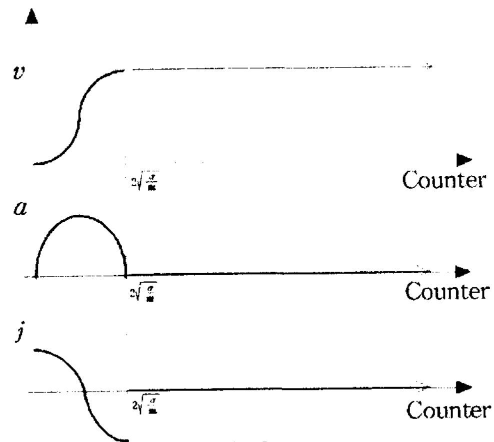

Fig.3.4

또한, $f(x)$는 분수 함수와 같이 일반 함수의 곱으로 이루어서 앞에서 언급한 일련의 조건을 만족시킬 수 있어야 한다.

우리는 식(3.3)의 2차 도함수 즉, 저크 함수 $j(x)$가 연속적인 특징을 만족시키기 위한 임의의

함수 $f(x)$를 결정한다는 것은 어렵고도 힘든 작업일 것이다. 따라서, 우리는 미분 기하학의 기초적인 원리를 배경으로 하여서, 함수 $f(x)$를 일단 예측할 필요가 있다. 분수 함수는 단순 조화 함수와 같이 초월 함수에 속하므로 앞에서의 분수 함수의 그래프 계형 Fig3.2의 형태는 Sine함수와 유사성을 띈다. 우리는 단순 조화 함수만을 가지고는 jerk함수의 연속성을 만족시키지 못하기 때문에 일반 함수와의 합성을 통해서 사이클로이드 변위 함수를 만들게 되는데, 마찬가지로 식 (3.2-1)에서 $f(x)$는 일반 함수의 곱으로 예측해 볼 수 있다.

3.4 함수 $v(x)=\dfrac{nx}{mx^2+a}+f(x)$ 의 예측

우선, 함수 $f(x)$를 예측하여 $f(x)=\beta x$로 놓고 이 속도 함수($v(x)$)의 계형의 극한값을 고찰하면 변수 x를 $\pm\infty$로 취했을 때, 양 끝단이 기울기 β에 수렴함으로 $\lim_{x\to\pm\infty} v(x)$의 점근선이 기울기 β임을 알 수 있다.

$$\begin{aligned}\lim_{x\to\pm\infty}&\ \frac{nx}{mx^2+a}+f(x)\\ &=\lim_{x\to\pm\infty}\frac{\frac{n}{x}}{m+\frac{a}{x^2}}+f(\pm\infty) \qquad (3.4)\\ &=f(\pm\infty)=\beta(\pm\infty)\end{aligned}$$

따라서 속도 함수 $v(x)$가 임의의 함수 $f(x)$를 더하기 전의 그래프 개형이 Fig3.1에서의 좌우 극한값이 0으로 수렴한 것에 비해서 $f(x)$를 더한 후, 기울기 β에 수렴함으로 β값의 조작에 의해서 Fig3.1에서 분리된 극점과 변곡점을 하나로 일치시킬 수 있음을 예측할 수 있다.

3.5 함수 $f(x)$의 결정

$f(x)$의 함수를 가장 단순한 일차 함수를 선택하였기 때문에 속도 함수($v(x)$)의 2차 도함수인 저크($j(x)$)함수에서는 $f(x)$가 소멸되기 때문에 저크 함수를 Zero 로 만드는 x값은 변함없이 $x=0,\pm\sqrt{\frac{3a}{m}}$ 에서 변곡점을 가진다.

$$v(x)=\frac{nx}{mx^2+a}+\beta x \qquad (3.5\text{-}1)$$

$$a(x)=\frac{an-mnx^2}{(mx^2+a)^2}+\beta=v'(x) \qquad (3.5\text{-}2)$$

$$j(x)=\frac{2mnx(mx^2-3a)}{(mx^2+a)^3}=v''(x) \qquad (3.5\text{-}3)$$

여기서 변곡점과 극값을 일치시키기 위해서는 $a(x)$함수가 $a(x=+\sqrt{\frac{3a}{m}})=0$ 이 되어야 한다. 따라서, 이때의 가속도 함수의 기울기 β는 $\beta=\dfrac{n}{8a}$인 값을 가진다.

결론적으로 우리가 구하고자 하는 최종 속도 함수 $v(x)$와 가속도 함수 $a(x)$ 그리고, Jerk 함수 $j(x)$는 다음과 같으며 이것을 이용하여 가속 구간에서 등속 구간으로 진입할 때의 V-A-J선도는 Fig 3.5와 같다.

$$v(x)=\frac{nx}{mx^2+a}+\frac{n}{8a}x \qquad (3.5\text{-}4)$$

$$a(x)=\frac{n(\sqrt{m}x-\sqrt{3}a)^2(\sqrt{m}x+\sqrt{3}a)^2}{8a(mx^2+a)^2}=v'(x) \qquad (3.5\text{-}5)$$

$$j(x)=\frac{2mnx(mx^2-3a)}{(mx^2+a)^3}=v''(x) \qquad (3.5\text{-}6)$$

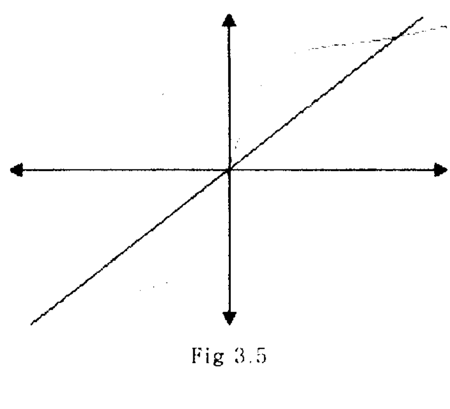

Fig 3.5

4. 속도 함수 $v(x)=\dfrac{nx}{mx^2+a}+\dfrac{n}{8a}x$ 의 활용

본 논문에서 도출한 이 속도 함수는 그 구성에서 알 수 있듯, 일반 제어 이론에서 알려진 함수들에 비해서 최소한의 곱셈 연산과 나눗셈 연산으로 CPU은 연산 처리 속도 부담을 줄일 수 있고, 또한 역학적인 안전의 기본 법칙에 위배되지

않는 가속도 table을 산출 할 수 있다. 앞에의 수식 3.5-4 에서 볼 수 있듯이 이 함수가 산출 할 수 있는 최대한의 가속도의 크기는 $x=0$ 일 때인데, 그 크기는 $a(0)=\frac{9n}{8a}$ 이고, 변수 m은 가속도의 크기의 변화에 관여하지 않으므로 1로 놓을 수 있다. 따라서, 가속하려는 구간의 거리 Counter는 Fig 3.6에서의 속도 그래프에서 볼 수 있듯이 가 구간 이 $\sqrt{3a}$ 이고 증가 속도는 $3n\sqrt{3a}/4a$ 인 크기를 가진다.

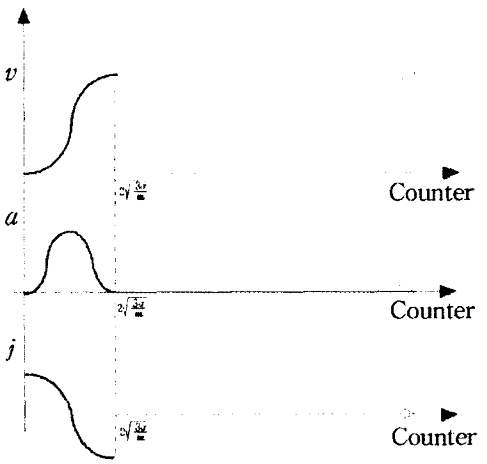

Fig 3.6

5. 결 론

지금까지 임의의 물성체를 제어하기 위해서 요구되는 가·감속의 제어 이론을 역학적 해석뿐만 아니라, 이것을 CPU의 연산 처리 속도에 부담이 적어서 실시간에 구현 할 수 있는 가·감속 제어 함수를 소개하였다. 이 속도 함수는 우리가 범용적으로 사용하고 있는 일반 함수의 곱인 4567-Polynomial. 이나 다른 삼각 함수의 곱, Simple harmonic function 같은 함수에 비해서 다음 세 가지의 두드러진 장점을 가진다.

첫째. 프로그램 상의 구현이 쉽고, 더욱이 어셈블리로 구현이 용이하기 때문에 그 연산 처리 속도가 타 함수에 비교할 수 없을 정도로 빠르다.

둘째. 이 분수 함수로 가·감속 구간의 다양한 선택에 있어서 그 조작이 용이하고, 도달하고자 하는 속도의 산출 또한 매우 쉽다.

셋째. 이것을 구현했을 때, 차지하는 프로그램은 다른 함수에 비하여 매우 적어므로 소형의 모터 제어 컨트롤러에 사용하기 적합하다.

우리는 이러한 이론의 실험의 적용에 있어서 스테핑 모터를 이용해서 실험하였으며. 그 실험 내용은 짧은 가속 구간으로 원하는 속도에 도달하는 것과 그리고, 모터의 탈조 영역에 접근하였을 때의 고속 제어에 중점을 두었으며, 이 모든 것은 어셈블리어로 작성하였고, 초고속의 실험에 있어 만족할 만한 성과를 보았다.

앞으로의 과제는, DC 모터 제이에 중점을 두고 좀더 범용성 있는 개념으로 확장해 나아갈 필요가 있고, 또한 기존에 알려져 있는 여러 Jerk 개념을 만족하는 수학 함수들과 비교하여 그 장단점을 좀 더 깊게 알아 볼 필요가 있다. 그리고, 이 분수 함수를 가·감속 개념의 도입에 머물지 않고, CAM설계에도 적용해 보아야 할 것이다.

후기

본 연구는 한국 과학 재단 지정 창원대학교 공작 기계 기술 연구 센터의 지원에 의한 것입니다.

참고문헌

1. J.L. Meriam, L.G. Kraige, *Engineering Mechanics Dynamics third edition*. John Wiley & Sons. Inc., 1993

2. Bela I. Sandor, *Engineering Mechanics Dynamics Second edition*. Prentice-Hall. Inc., 1987

3. Robert L. Norton, *Design of Machinery*. MCGRAW-HILL International Editions. 1999

4. Charles E. Wilson/J. Peter Sadler. *Kinematics and Dynamics of Machinery*.