Journal of Institute of Control, Robotics and Systems (2010) 16(10):997-1005
DOI:10.5302/J.ICROS.2010.16.10.997 ISSN:1976-5622

시스템 사양을 고려한 컨볼루션 기반 목표궤적 생성 방법

Convolution-based Desired Trajectory Generation Method Considering System Specifications

이 건, 최 영 진*, 김 진 현
(Geon Lee[1], Youngjin Choi[1], and Jinhyun Kim[2])
[1]Hanyang University
[2]Seoul National University of Science and Technology

Abstract: Most motion control systems consist of a desired trajectory generator, a motion controller such as a conventional PID controller, and a plant to be controlled. The desired trajectory generator as well as the motion controller is very important to achieve a good tracking performance. Especially, if the desired trajectory is generated actively utilizing the maximum velocity, acceleration, jerk and snap as given system specifications, the tracking performance would be better. For this, we make use of the properties of convolution operator in order to generate a smooth (S-curve) trajectory satisfying the system specifications. Also, the proposed trajectory generation method is extended to more general cases with arbitrary initial and terminal conditions. In addition, the suggested trajectory generator can be easily realized for real-time implementation. Finally, the effectiveness of the suggested method is shown through numerical simulations.

Keywords: trajectory generation, jerk-bounded, system specifications, convolution operator

I. 서론

로봇의 동작 제어시스템은 크게 목표 궤적을 생성하는 궤적 생성기, 생성된 궤적 추종을 달성하게 하는 제어기, 구동기를 포함하는 제어 대상시스템으로 나뉠 수 있다. 제어기는 생성된 목표 궤적을 추종하면서 발생되는 오차를 빠르고 정확하게 보상해주는 역할을 하며, 궤적 생성기는 목표 작업 공간에서의 작업 궤적을 생성해주는 역할을 담당한다. 따라서 제어 시스템의 높은 성능을 위해서는 추종 제어기뿐만 아니라 궤적 생성기 역시 매우 중요한 역할을 한다.

궤적 생성에 있어서 가장 중요시 되는 것은 속도나 가속도까지 미분 가능한 S-curve 특성의 위치 궤적을 생성하는 것이다. 특히, 가속도의 미분인 저크(jerk)가 상한/하한의 경계 이내에서 생성되는 궤적은 떨림 현상으로 인한 로봇의 손상을 줄여주며, 궤적 추종 정확도와 추종 속도를 향상시킨다[1]. 만약 대상 제어 시스템의 특성 중의 하나인 저크가 작게 제한되어 있다면, 작은 값으로 제한된 저크로부터 생성된 궤적은 동작 시간이 다소 증가하지만, 대상 제어시스템을 포함한 전체 제어시스템으로 하여금 더 빠르고 정확한 추종을 가능하게 해주며, 보다 부드러운 동작 구현을 가능하게 한다[2]. 예를 들어, 인간과 접촉을 하는 로봇이나 정밀한 제어를 요하는 디스크 드라이브의 경우, 저크가 작은 값으로 제한된 궤적이 요구된다[3]. 각 로봇은 내장된 구동기의 최대 속도 및 가속도와 같은 최대 성능이 각기 다르므로 높은 성능의 동작 제어를 위해서는 목표 궤적 생성시 구동기의 최대 성능 또한 고려되어야 한다[2,4]. 구동기의 최대 성능을 넘어서 생성되는 궤적은 추종이 불가능 할 뿐만 아니라 과부하로 인한 제어 시스템의 손상을 유발 할 수 있다. 또한, 이러한 궤적 생성기 사용의 경우, 실험적으로 추종 가능한 궤적의 범위를 찾아야 하므로 많은 시간과 노력이 필요하게 된다. 산업용 로봇 적용에 있어서는 경제적 비용의 이유로 적은 계산 량 역시 고려되어야 하는 중요한 부분 중 하나이다. 적은 계산량은 처리 속도가 빠르지 않은 저가의 프로세서에서의 실시간 동작 제어를 가능하게 하므로 경제적 비용을 줄일 수 있다[5]. 또한 실시간으로 동작하는 시스템에서의 제어를 가능하게 해준다[6].

로봇에서의 궤적 생성을 위해, 일반적으로 시간에 대한 3차 이상의 다항식(polynomial)을 이용한 방법이 많이 쓰인다[7]. 다항식을 이용한 방법은 차수에 따라 속도 및 가속도 또는 더 상위 미분 함수의 다양한 특성 설정이 가능하지만 그 값에 제한을 주기 매우 어렵다는 단점이 있다. 고차 다항식을 이용하여 궤적 생성시 동작 제어 시스템의 성능 제한을 반영 할 경우, 궤적 생성을 위한 계산 량은 크게 증가하게 된다. 참고문헌[8-11]에서는 적은 계산 량의 증가로 다항식을 이용하여 시스템의 성능 제한을 반영할 수 있는 궤적 생성 방법을 제시하였다. 또한, 증가하는 계산 량을 줄이기 위해 다항식을 변형한 ASPOT (Arbitrary States POlynomial-like Trajectory)의 방법도 제안되었다[5]. 동작 제어 시스템 성능 제한을 반영하는 또 다른 방법으로는 LSPB (Linear Segment with Parabolic Blends)를 이용한 궤적 생성 방법이 있다[2,12]. LSPB의 방법은 Bang-Singular-Bang의 방법으로 구동기 성능이 고려된 최적화된 시간(time-optimal)의 궤적 생성 방법이지만, 이런 최적화 방법들은 그에 상응하는 계산 량을 필요로 하므로 실시간 제어를 위해서는 계산 량에 맞는 경제적 비용

* 책임저자(Corresponding Author)
논문접수: 2010. 3. 29., 수정: 2010. 7. 19., 채택확정: 2010. 8. 6.
이건, 최영진: 한양대학교 대학원 전자전기제어계측공학과
(emerald@hanyang.ac.kr/cyj@hanyang.ac.kr)
김진현: 서울과학기술대학교 기계공학과(jinhyun@snut.ac.kr)
※ 이 논문은 교육과학기술부의 재원으로 한국연구재단의 중견연구자지원사업(No. 2008-0061778)과 지식경제부의 전략기술인력양성사업 및 융복합형로봇전문인력양성사업에 의하여 연구 되었음.

Copyright© ICROS 2010

을 지불해야 한다. [6]에서는 기존의 방법 보다 더 적은 계산량으로 최적화된 시간의 궤적 생성이 가능한 방법을 제안하였다.

위에 기술된 다항식을 기본으로 하는 궤적 생성과 다른 방법으로 디지털 컨볼루션(digital convolution)을 이용한 방법이 있다[4,13]. 일반적인 연속시간 컨볼루션은 많은 계산 량을 필요로 하지만, 식의 변형을 통한 디지털 (이산시간) 컨볼루션의 방법은 다항식을 이용한 방법과 비교하였을 때, 비슷한 계산 량으로 S-curve 특성의 궤적 생성이 가능하다. 또한, 반복되는 형태의 계산으로 속도, 가속도 및 저크 등의 최대값을 제한시킬 수 있다는 장점이 있다. 하지만 기존의 컨볼루션 방법은 초기 및 최종 속도 설정이 불가능하기 때문에 다양한 로봇에 적용하기엔 무리가 있다. 예를 들어, 운동 제어 시스템이 다른 물체를 친다거나 공중으로 점프를 하는 경우에는 최종 속도 설정이 가능하여야 한다. 본 논문에서는 컨볼루션을 이용하여, 초기 및 최종 속도 설정이 가능한 궤적 생성 방법을 제안한다.

제안 하는 방법은 컨볼루션 되는 함수를 설정함으로써 속도, 가속도 또는 저크 등의 최대값을 특정 값으로 제한 할 수 있다. 이를 위해 본 논문에서는 가속도, 저크와 같은 속도의 미분 함수들을 n 차 속도 함수라 정의한다. 예를 들어, $n=1$ 일 경우는 1차 속도함수로서 가속도함수를, $n=2$ 일 경우는 2차 속도함수로서 저크함수를, $n=3$ 일 경우는 3차 속도함수로 저크함수의 미분인 스냅(Snap)함수를 의미한다. 또한 제안 하는 방법은 기존의 컨볼루션 방법의 특성을 그대로 가지고 있어서 적은 계산 량으로 궤적 생성 연산이 가능하며 n 번 반복적인 계산으로 n 차 속도 함수까지 특정 값으로 제한이 가능하다.

본 논문은 다음과 같이 구성된다. 먼저, 제안하는 궤적 생성 방법에 이용할 컨볼루션의 특성에 대해 간략히 설명한다. 이를 바탕으로 컨볼루션 함수와 n 차 속도 함수의 최대값에 대한 관계를 이끌어내며, 그 의미를 확장시켜 초기 및 최종 속도 결정이 가능한 새로운 궤적 생성 방법을 제안한다. 또한, 컨볼루션의 복잡한 계산식을 간단한 순차방정식 형태로 변환해주는 디지털 컨볼루션 기법에 대해 기술하며, 최종적으로 목표 궤적 생성기의 전 과정을 정리한다. 마지막으로 모의실험을 통해 제안하는 방법의 유효성 및 적합성을 보이며 본 논문의 결론을 기술한다.

II. 컨볼루션(convolution)의 기법

컨볼루션은 사각파와 같은 불연속 함수를 연속 함수로 변화시킬 수 있다. 더욱이 반복적인 컨볼루션의 적용을 통해 미분 가능한 S-curve의 함수까지도 얻을 수 있다. 예를 들어, 다음 식과 같이 함수 $y_0(t)$ 를 구간 $0 \le t \le t_0$ 에서 v_0 의 값을 갖는 사각파형의 함수라 가정해 보자.

$$y_0(t) = \begin{cases} v_0, & 0 \le t \le t_0 \\ 0, & otherwise \end{cases} \tag{1}$$

또한 n 번째 적용되는 컨볼루션 함수 $h_n(t)$는 다음 식과 같이 $0 \le t \le t_n$ 의 구간에서 단위 면적(unit area)을 갖는 사각파형의 함수라 가정한다.

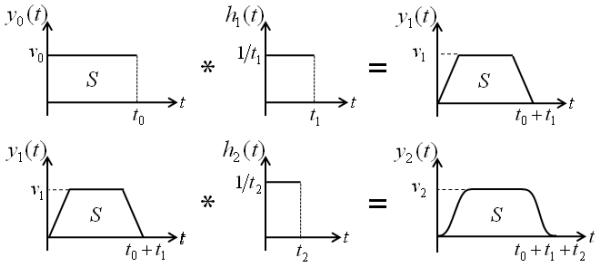

그림 1. 사각파형 함수에 대한 두 번의 컨볼루션 결과.
Fig. 1. Result of twice convolutions for rectangular function.

$$h_n(t) = \begin{cases} 1/t_n, & 0 \le t \le t_n \\ 0, & otherwise \end{cases} \tag{2}$$

여기서 함수 $y_n(t)$를 n 번의 컨볼루션이 적용된 결과 함수라 하면, $y_0(t)$ 와 $h_1(t)$ 의 컨볼루션의 결과는 $y_1(t)$ 로 나타낼 수 있으며, $y_2(t)$ 는 $y_1(t)$ 와 $h_2(t)$ 의 컨볼루션의 결과를 나타낸다. 연속적인 컨볼루션의 결과 함수 $y_1(t)$ 와 $y_2(t)$ 는 그림 1과 같이 나타내어진다.

그림 1에서 볼 수 있듯이, 한 번의 컨볼루션은 사다리꼴형의 함수를 생성한다. 한번의 컨볼루션 적용으로 얻어진 $y_1(t)$ 함수의 미분 함수는 사각파형으로 이루어진 함수이며, 이는 그 최대값이 특정 값에 제한이 되어 있음을 의미한다. 두 번의 컨볼루션은 S-curve 곡선의 함수를 생성하며, 두 번의 컨볼루션의 적용으로 얻어진 $y_2(t)$ 함수의 미분 함수는 사다리꼴로 이루어진 함수가 된다. 따라서 두 번 컨볼루션의 결과인 $y_2(t)$ 의 두 번 미분 함수는 사각파형으로 이루어진 함수이며, 이 함수의 두 번 미분 함수의 최대값 또한 특정 값에 제한이 되어 있음을 의미한다. 이처럼 컨볼루션 횟수 n 이 증가함에 따라, 결과 함수는 더욱 부드러운 S-curve 곡선으로 형성되며, 결과 함수의 n 번 미분 함수는 특정 값에 제한된다.

만약 초기 $y_0(t)$ 가 사각파형 속도함수로 주어지고, n 번의 컨볼루션을 적용함에 의해 결과 속도함수 $y_n(t)$을 얻는다면, 결과 속도함수 $y_n(t)$의 최대 속도 값이 어느 특정 값으로 제한되며, 이를 한번 미분하여 얻은 1차 속도함수(가속도 함수)도 최대값이 특정 값으로 제한되며, 이를 일반화하여 n 번 미분하여 얻은 n 차 속도함수의 최대값 또한 특정 값으로 제한된다. 이외에도 사각파형의 함수에 연속적인 컨볼루션을 적용함에 의하여 얻어진 결과 함수는 다음과 같은 몇 가지 특성을 만족한다.

특성 1: n 번 컨볼루션을 적용하여 얻어진 결과 함수 $y_n(t)$는 초기 $y_0(t)$가 0이 아닌 값으로 정의되는 구간 t_0 부터 $h_n(t)$의 구간 t_n 까지의 합인 구간 $0 \le t \le \sum_{k=0}^{n} t_k$ 에서 값이 정의된다.

특성 2: 적용되는 컨볼루션 횟수와는 무관하게, 초기의 함수 $y_0(t)$의 넓이는 n 번 컨볼루션을 적용하여 얻어진 결과 함수 $y_n(t)$의 넓이와 항상 같다.

$$\int_0^{t_0} y_0(t)\ \ dt = \int_0^{\sum_{k=0}^{n} t_k} y_n(t)\ \ dt \tag{3}$$

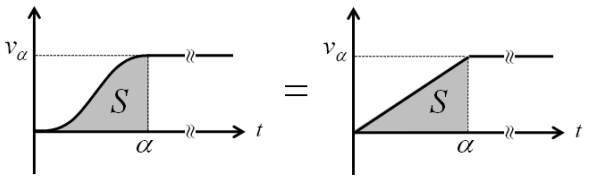

그림 2. 컨볼루션 결과 함수의 이동 거리.
Fig. 2. Moving distance of result function obtained after performing convolution.

특성 3: 초기의 함수 $y_0(t)$의 최대값 v_0은 n 번 컨볼루션을 적용하여 얻어진 결과 함수 $y_n(t)$의 최대값 v_n 보다 항상 크거나 같다.

$$\max(|y_0(t)|) \geq \max(|y_n(t)|) \tag{4}$$

특성 4: 컨볼루션 적용 후 얻어진 결과 함수가 임의의 속도 변화 후 그 값이 유지 될 경우, 변속 시 가속도 및 감속도 특성은 대칭으로 생성된다. 따라서 그림 2와 같이 변속 동안의 이동 거리는 삼각형의 넓이와 같다.

III. 궤적 생성 방법

본 장에서는 먼저 두 번의 연속적인 컨볼루션 적용 예를 통해 제안하고자 하는 목표 궤적생성 방법을 설명한 후 이를 일반화 시킨다. 또한 본 논문에서 제안되는 궤적생성 방법의 쉬운 설명을 위해 운동제어 시스템은 단축상의 점과 점 사이를 움직이는 시스템이라 가정하며, 제어 대상 운동시스템은 $v_{\max}$, $a_{\max}$, $j_{\max}$ 로 표현되는 각각의 최대 속도, 최대 가속도, 최대 저크를 갖는다고 가정한다.

1. 컨볼루션 적용 결과함수와 n 차 속도함수와의 관계

운동제어 시스템이 최대속도 $v_{\max}$ 로 주어진 이동거리 S 를 움직인다고 가정하면, 이 시스템의 속도-시간 함수는 식 (1)과 같은 사각파형의 속도함수로 표현 가능하며, 이 때의 v_0과 t_0 는 다음 식과 같다.

$$v_0 = v_{\max}, \quad t_0 = \frac{|S|}{v_{\max}} \tag{5}$$

II 장에서 제시되었듯이, 위의 사각파형의 초기 속도함수에 두 번의 컨볼루션을 적용하면 최대값이 특정 값에 제한되는 S-curve 곡선의 결과 속도함수를 얻을 수 있다. 또한 얻어진 결과 속도함수는 두 번까지 미분가능하며, 한 번 미분하여 얻은 1차 속도함수(가속도함수)와 한번 더 미분하여 얻은 2차 속도함수(저크함수)도 최대값이 특정 값에 의해 제한된다. 그리고 이동거리인 결과 속도함수의 넓이는 특성 2에 의해 컨볼루션 횟수와 관계 없이 항상 같게 되며, 특성 3에 따라 결과 속도함수의 최대값은 $v_{\max}$ 이하로 생성된다. 이 경우, 컨볼루션의 적용 구간들이 $t_0 \geq t_1 + t_2$ 와 $t_1 \geq t_2$ 의 두 조건을 만족하게 되면 결과 속도함수의 최대값은 $v_{\max}$ 와 같게 되며, 그 이외의 경우에서는 결과 속도함수의 최대값이 $v_{\max}$ 보다 항상 작게 생성된다. 초기의 속도함수의 최대값 v_0와 n 번의 컨볼루션의 적용을 통해 얻어지는 결과 속도함수의 최대값 v_n는 일반적으로 다음 식과 같은 관계를 만족한다.

$$\begin{cases} v_0 = v_n, & for\ \ t_{i-1} \geq \sum_{k=i}^{n} t_k \,(i = 1,\ 2, \cdots) \\ v_0 > v_n, & otherwise \end{cases} \tag{6}$$

앞 장에서 우리는 n 차 속도함수를 결과 속도함수에 대한 n 번 미분함수로 정의하였다. 여기서 n 차 속도함수들의 최대값을 활용하기 위해서 컨볼루션 되는 각 사각파형의 함수의 정의구간을 결정하는 t_n 은 다음 관계를 만족한다.

$$t_n = \frac{v_{\max}^{(n-1)}}{v_{\max}^{(n)}} \quad i = 1,\ 2, \cdots \tag{7}$$

위의 식에서 $v_{\max}^{(n)}$ 는 n 차 속도함수의 최대값을 의미하며, 일반적으로 양수이다. 예를 들어, $v_{\max}^{(1)}$ 는 1차 속도함수(가속도함수)의 최대값인 $a_{\max}$ 를 의미하며, $v_{\max}^{(2)}$ 는 2차 속도함수(저크함수)의 최대값인 $j_{\max}$ 를 의미한다.

컨볼루션을 이용한 S-curve 형태의 결과 속도함수 생성방법은 다음의 과정을 따른다. 첫째, 주어진 이동거리와 최대속도로부터 식 (5)를 이용하여 그림 1과 같은 사각파형 초기 속도함수의 v_0와 t_0를 결정한다. 둘째, 주어진 제어 대상 운동 시스템의 성능특성 중 식 (7)로부터 최대 속도에 대한 최대 가속도의 비율로 얻어지는 t_1 을 결정하고 이를 이용하여 그림 1과 같이 컨볼루션을 수행하여 사다리꼴형의 속도함수를 얻을 수 있다. 셋째, 식 (7)을 한번 더 이용하여 최대 가속도에 대한 최대 저크의 비율로 얻어지는 t_2 를 결정하고 컨볼루션을 한번 더 수행하여 그림1과 같은 S-curve 형태의 결과 속도함수를 얻을 수 있다. 이를 일반화하여 식 (7)로부터 t_n 을 결정하고 이를 이용하여 만들어진 단위 면적의 사각파형 함수를 반복적으로 컨볼루션 연산을 수행함으로써 n 차 속도함수의 최대값이 특정 값으로 제한되는 목표궤적을 생성할 수 있다.

초기 속도 및 최종 속도가 '0'이 아닐 경우, 가속/감속 구간에서의 속도 변화는 항상 '0'과 $v_{\max}$ 사이에 존재하지 않기 때문에, 최대 가속도 활용을 위한 t_1 은 식 (7)과 다르게 된다. 다음 절에서는 '0'이 아닌 초기 및 최종 속도의 경우의 S-curve 속도함수 생성을 위한 v_0, t_0, t_1 를 결정하는 방법에 대해 설명한다.

2. '0'이 아닌 최종 속도

그림 3에서와 같이, 시간 t_0까지 v_0의 값을 갖고 그 후로 $\sum_{k=0}^{n} t_k$ 이상의 임의의 시간 α 동안 v_f 의 값을 갖는 계단파형의 속도 함수를 생각해보자. 이 계단파형의 초기 속도함수를 단위 면적의 사각파형 함수로 두 번의 컨볼루션을 수행하면 S-curve의 결과 속도함수를 얻을 수 있다. 그림 3에서 볼 수 있듯이, 컨볼루션의 특성상 시간이 흐름에 따라 함수의 최종 값은 '0'으로 수렴하지만 컨볼루션 횟수 n 에 따라 속도 함수는 $\sum_{k=0}^{n} t_k$ 의 시간에서 v_f 의 값을 갖는다. 또한, 특성 2에 의하여 그림 3에 제시된 각 속도함수들의 전체 넓이는 모두 같으며, 특성 4를 이용하여 n 번 컨볼루션을 수행한 결과 속도함수에서 시간 $\sum_{k=0}^{n} t_k$ 까지의 넓이 S_n 와 계단

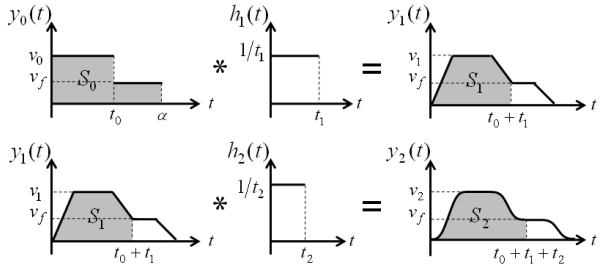

그림 3. 계단 파형의 함수와 컨볼루션 수행 후의 결과 함수.
Fig. 3. Result of twice convolutions for step-wise functions.

파형 함수의 전체 넓이 S_0 는 다음 식과 같은 관계를 갖는다.

$$\begin{aligned} S_n &= S_0 - \frac{v_f}{2}\sum_{k=1}^{n} t_k - v_f\left(\alpha - \sum_{k=0}^{n} t_k\right) \\ &= v_0 t_0 + v_f(\alpha - t_0) - \frac{v_f}{2}\sum_{k=1}^{n} t_k - v_f\left(\alpha - \sum_{k=0}^{n} t_k\right) \\ &= v_0 t_0 + \frac{v_f}{2}\sum_{k=1}^{n} t_k \end{aligned} \tag{8}$$

위에서 설명한 특성을 이용하여 '0'이 아닌 최종 속도의 궤적은 계단파형 함수를 특정 시간까지 컨볼루션 함으로써 생성할 수 있다. '0'이 아닌 최종 속도의 궤적의 경우, v_0 과 v_f 의 값에 따라 최대 가속도 활용을 위한 t_1 은 결정된다. 최소 이동시간을 위해서 v_0 는 $v_{\max}$, $-v_{\max}$ 중의 한가지 값을 취할 수 있으며, v_0 값의 선택 방법에 대해서는 뒤에서 다시 설명하도록 한다. 또한, v_0 가 '0'의 값을 취할 경우, t_1 은 $|v_f|/a_{\max}$ 이 되며 t_0 는 무의미한 값이 되므로 이 경우 최소 이동시간을 위해서는 t_0 는 '0'이어야 한다.

v_0 값의 결정에 앞서, 먼저 그림 4과 같이 t_0 가 '0'의 값을 갖는 경우를 생각해 보자. 이 경우, 그림 4(a)와 같이 속도 함수는 최종 속도 v_f 로 이루어진 사각파형 함수가 되며, 이 사각파형 함수를 두 번 컨볼루션을 수행함으로써 그림 4(b)와 같은 s-curve 형태의 속도 궤적 함수를 얻을 수 있다. 그림 4(b)에서 어두운 영역으로 표시된 시간 t_1+t_2 까지의 넓이 S^* 는 특성 4에 의해 사각파형 함수 넓이의 반이 된다. t_1 이 최대 가속도 활용을 만족할 때, 즉 t_1 이 $|v_f|/a_{\max}$ 일 때의 S^* 의 넓이를 S_f^* 라 정의하면, 주어진 이동 거리 S 가 S_f^* 의 넓이와 같을 경우 t_0 는 '0'이 된다. 이와 같은 경우, 시간

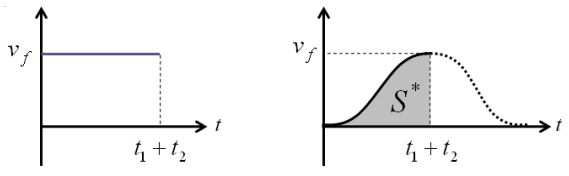

그림 4. 최종 속도 v_f 을 갖는 사각파형과 S-curve 속도 함수.
Fig. 4. Rectangular velocity profile and S-curve velocity profile satisfying a final velocity v_f.

t_1+t_2 은 시스템이 최대 가속도와 최대 저크를 가지고 속도 '0'에서 v_f 까지 변속하기 위해 걸리는 최소 시간이며, S_f^* 는 그 최소 시간 동안 변속하며 이동하게 되는 최소 거리를 의미한다. 이를 n 번 컨볼루션을 적용한 것으로 일반화한 S_f^* 는 컨볼루션의 횟수 n 에 따라 다음 식과 같이 표현된다.

$$S_f^* = \frac{v_f}{2}\left(\frac{|v_f|}{a_{\max}} + \sum_{k=2}^{n} t_k\right) \tag{9}$$

다음으로 t_0가 '0'이 아닌 값을 가질 경우를 생각해보자. v_0 와 v_f 의 부호가 같을 경우, 가속/감속 구간 동안 최대 속도 변화는 $v_{\max}$ 가 되며, 따라서 최대 가속도를 위한 t_1 는 $v_{\max}/a_{\max}$ 가 된다. v_0와 v_f 의 부호가 반대일 경우, 가속/감속 구간 동안 최대 속도 변화는 $v_{\max}+|v_f|$ 가 되므로 최대 가속도를 위한 t_1 은 $(v_{\max}+|v_f|)/a_{\max}$ 가 된다. 만약 식 (8)에서 t_0 가 무한히 '0'에 가까워지면 이동거리 S^*는 $\left(v_f\sum_{k=1}^{n} t_k\right)/2$ 가 되며, t_1 이 $v_{\max}/a_{\max}$ 일 때 최소 이동거리를 갖는다. 이때의 최소 이동거리를 S_m^* 라 정의하면 S_m^* 는 다음과 같다.

$$S_m^* = \frac{v_f}{2}\left(\frac{v_{\max}}{a_{\max}} + \sum_{k=2}^{n} t_k\right) \tag{10}$$

일반적으로 v_0가 '0'이 아닌 값을 가질 때는 t_1이 $v_{\max}/a_{\max}$ 보다 크므로, 주어진 이동거리는 식 (10)의 S_m^* 보다 크거나 작게 된다. 따라서, 주어진 목표 이동거리 S 가 S_m^* 보다 클 경우 v_0 은 $v_{\max}$ 의 값을 취하게 되며, S_m^* 보다 작을 경우 v_0 은 $-v_{\max}$ 의 값을 취하게 된다. S 가 S_m^* 와 같을 경우는 최대 가속도 활용을 위한 궤적 생성이 불가능하며, 이 경우 본 논문에서는 최소 이동 시간을 위해 t_0는 '0', t_1은 $v_{\max}/a_{\max}$ 로 설정하도록 한다. 주어진 이동 거리 S 가 S_f^* 또는 S_m^* 과 같을 경우, 즉 t_0가 '0'의 값을 갖게 될 경우, 식 (8)에서 볼 수 있듯이 v_0는 이동거리 에 아무런 영향을 미치지 않는 무의미한 값이 된다. 위에서 설명한 것과 같이, v_0와 t_1은 주어진 이동거리 S 와 S_f^*, S_m^* 의 값의 관계에 의해서 다음과 같은 식들로 정리할 수 있다.

$$v_0 = \left|\ \mathrm{sgn}(S - S_f^*)\right|\mathrm{sgn}(S - S_m^*)v_{\max} \tag{11}$$

$$t_1 = \frac{\left|\mathrm{sgn}(S - S_m^*)\right|\max\left(|v_0|,\ |v_0 - v_f|\right) + \left(1 - \left|\mathrm{sgn}(S - S_m^*)\right|\right)v_{\max}}{a_{\max}} \tag{12}$$

또한 식 (11)를 식 (8)에 대입하여 풀면 미지의 수 t_0는 다음과 같이 구해진다.

$$t_0 = \frac{S - \frac{v_f}{2}\sum_{k=1}^{n} t_k}{\left|\mathrm{sgn}(S - S_f^*)\right|\mathrm{sgn}(S - S_m^*)v_{\max}} \tag{13}$$

위의 식은 S 에 따라 부정의 해를 갖는다. 본 논문에서는

식의 변형을 통해 이 문제를 해결하며, 식의 변형 방법은 다음 절에서 언급하기로 한다.

마지막으로 연속적인 컨볼루션 적용을 위한 사각파형 함수 결정을 위한 t_2부터 t_n까지의 정의구간 계수는 속도와 무관함으로 식 (7)을 이용하여 결정된다.

3. '0'이 아닌 초기 속도와 최종 속도

본 절에서는 궤적 생성시 초기 속도 v_i가 추가 됨으로써 발생하는 컨볼루션 초기 함수 $y_0(t)$의 변화에 대해 설명한다. 그림 5에서 볼 수 있듯이, '0'이 아닌 초기 속도 궤적은 시작점이 '0'인 S-curve의 함수와 초기 속도의 크기를 갖는 사각파형의 함수로 분해될 수 있다. 여기서, S-curve 함수는 앞 절에서 언급했던 것과 같은 방법으로 계단파형의 함수에 두 번의 컨볼루션을 수행하여 생성이 가능하며, 생성된 S-curve 함수에 초기 속도 v_i를 더해 줌으로써 '0'이 아닌 초기 속도의 궤적 생성이 가능하다. 그림 5에서 v_f' 은 실제 최종 속도 v_f 와 초기 속도 v_i 의 차인 $v_f - v_i$ 가 되며, v_n' 은 $v_n - v_i$ 이 된다. 또한, 컨볼루션 횟수 n 에 따른 함수의 넓이 S_n 은 S_n^t 와 S_n^b 의 합으로 다음과 같이 표현된다.

$$\begin{aligned} S_n &= v_0' t_0 + \frac{v_f - v_i}{2}\sum_{k=1}^{n} t_k + v_i \sum_{k=0}^{n} t_k \\ &= (v_0' + v_i)t_0 + \frac{v_f + v_i}{2}\sum_{k=1}^{n} t_k \\ &= v_0 t_0 + \frac{v_f + v_i}{2}\sum_{k=1}^{n} t_k \end{aligned} \tag{14}$$

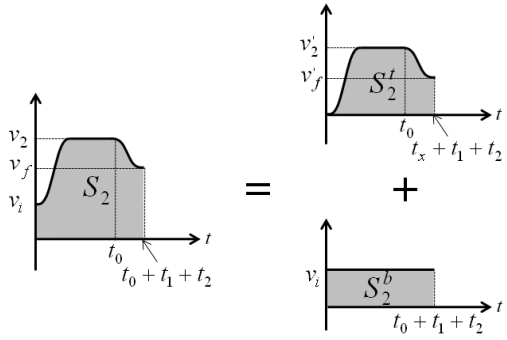

그림 5. 주어진 거리를 이동하는 속도 함수의 합성.

Fig. 5. Composition of velocity function profiles moving given distance.

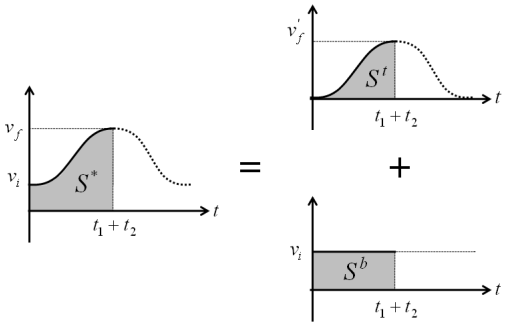

그림 6. 주어진 거리를 이동하는 속도 함수의 합성.

Fig. 6. Composition of velocity function moving given distance.

t_0 가 '0'인 경우 또한, 그림 6와 같이 두 개의 속도 함수로 분해 될 수 있다. 최대 가속도를 만족하기 위한 t_1 은 $|v_f - v_i|/a_{\max}$ 가 되며, 이 때의 최소 이동거리 S_f^* 는 S_f^t 과 S_f^b 의 합으로 표현된다. 따라서, 식 (9)은 다음과 같이 변형된다.

$$\begin{aligned} S_f^* &= \frac{v_f - v_i}{2}\left(\frac{|v_f - v_i|}{a_{\max}} + \sum_{k=2}^{n} t_k\right) + v_i\left(\frac{|v_f - v_i|}{a_{\max}} + \sum_{k=2}^{n} t_k\right) \\ &= \frac{v_f + v_i}{2}\left(\frac{|v_f - v_i|}{a_{\max}} + \sum_{k=2}^{n} t_k\right) \end{aligned} \tag{15}$$

비슷한 방법으로 v_0' 결정 기준이 되는 이동거리 S_m^* 는 S_m^t 와 S_m^b 의 합으로 표현된다. t_0 가 '0'이 아닐 경우, 최대 가속도 만족을 위한 시간 동안의 속도 변화량은 v_0', v_i, v_f 의 관계에 따라 계산된다. 최소 속도 변화량을 Γ 라 정의하면 Γ 는 식 (16)과 같이 나타내어지며 S_m^* 에 대한 식 (10)은 식 (17)과 같이 변형된다.

$$\Gamma = \min\left(v_{\max} - \min(v_i, v_f),\ v_{\max} + \max(v_i, v_f)\right) \tag{16}$$

$$S_m^* = \frac{v_f + v_i}{2}\left(\frac{\Gamma}{a_{\max}} + \sum_{k=2}^{n} t_k\right) \tag{17}$$

또한, n 번 컨볼루션 결과 함수의 최대값 v_n' 이 최대 속도를 넘지 않기 위해서 v_0' 는 초기 속도가 고려된 $(v_{\max} - v_i)$ 또는 $(v_{\max} + v_i)$ 의 값을 가져야 한다. 따라서, 주어진 이동거리 S 와 S_f^*, S_m^* 의 값의 관계에 의해 결정되는 v_0' 는 다음과 같이 나타내어진다.

$$v_0' = \left|\mathrm{sgn}(S - S_f^*)\right|\mathrm{sgn}(S - S_m^*)v_{\max} - v_i \tag{18}$$

v_0' 은 식 (18)과 같이, 주어진 목표 이동거리 S 가 S_m^* 보다 클 때 $v_{\max} - v_i$ 이 되며, S_m^* 보다 작을 때 $-v_{\max} - v_i$ 이 된다. 또한, S 가 S_m^* 나 S_f^* 와 같을 경우 t_0 는 '0'이 되므로 v_0' 는 결과 값에 어떠한 영향을 주지 않는다. v_0' 와 v_f' 의 관계에 따라 결정되는 t_1 에 대한 식 (12)은 식 (19)와 같이 변형되며, 식 (18)을 식 (14)에 대입하여 식 (20)의 t_0 를 구할 수 있다.

$$t_1 = \frac{\left|\mathrm{sgn}(S - S_m^*)\right|\max\left(\left|v_0'\right|, \left|v_0' - v_f'\right|\right) + \left(1 - \left|\mathrm{sgn}(S - S_m^*)\right|\right)\Gamma}{a_{\max}} \tag{19}$$

$$t_0 = \frac{S - \dfrac{v_f + v_i}{2}\displaystyle\sum_{k=1}^{n} t_k}{\left|\mathrm{sgn}(S - S_f^*)\right|\mathrm{sgn}(S - S_m^*)v_{\max}} \tag{20}$$

식 (20)의 경우, 주어지는 S 의 값에 따라 부정의 해를 가질 수 있다. 하지만 주어진 이동거리 S 가 S_f^* 또는 S_m^* 와 같을 경우, 즉 부정의 해를 갖게 되는 경우 t_0 는 '0'의 값을

가져야 한다. 또한, sgn(·) 이 단지 부호를 결정해주는 역할을 하는 것을 감안하여 식 (20)은 식 (21)처럼 변환이 가능하다.

$$t_0 = \frac{\left|\mathrm{sgn}(S - S_f^*)\right|\mathrm{sgn}(S - S_m^*)\left(S - \frac{v_f + v_i}{2}\sum_{k=1}^{n} t_k\right)}{v_{\max}} \tag{21}$$

IV. 디지털 컨볼루션으로의 변환

연속시간(continuous-time)에서의 컨볼루션은 많은 계산 량이 필요하므로 실시간(real-time)으로 동작하는 시스템 적용에 적합하지 않다. 하지만 이산시간(discrete-time)에서의 컨볼루션은 약간의 변화를 통해 적은 계산량의 컨볼루션 식으로의 표현이 가능하기 때문에, 제안된 방법의 실시간 동작 구현을 가능하게 해준다. 계산 량을 줄이기 위한 디지털 컨볼루션 형태는 다음과 같은 과정으로 생성된다.

임의의 함수 $y_{n-1}(t)$ 와 단일면적의 사각파형 함수 $h_n(t)$ 의 컨볼루션 결과로 생성되는 함수 $y_n(t)$ 는 샘플링 시간 T_s 를 이용하여 다음 식과 같이 이산시간에서의 컨볼루션으로 표현될 수 있다.

$$\begin{aligned} y_n(kT_s) &= \sum_{l=-\infty}^{\infty} h_n(kT_s)\, y_{n-1}((k-l)T_s) \\ &= \frac{1}{m_n}(y_{n-1}(kT_s) + y_{n-1}((k-1)T_s) \\ &\quad + y_{n-1}((K-2)T_s) + \cdots + y_{n-1}((k-m+1)T_s)) \end{aligned} \tag{22}$$

또한, 한 단계 전 시간의 결과 함수 $y_n((k-1)T_s)$ 는 위와 같은 방법으로 다음 식과 같이 전개된다.

$$\begin{aligned} y_n((k-1)T_s) &= \sum_{l=-\infty}^{\infty} h_n(kT_s)\, y_{n-1}((k-1-l)T_s) \\ &= \frac{1}{m_n}(y_{n-1}((k-1)T_s) + y_{n-1}((k-2)T_s) \\ &\quad + y_{n-1}((K-3)T_s) + \cdots + y_{n-1}((k-m)T_s)) \end{aligned} \tag{23}$$

식 (22)에서 식 (23)를 빼면, 두 번의 덧셈과 한번의 나눗셈으로 계산이 가능한 다음과 같은 디지털 컨볼루션 식을 얻을 수 있다.

$$y_n(kT_s) = \frac{y_{n-1}(kT_s) - y_{n-1}((k-m_n)T_s)}{m_n} + y_n((k-1)T_s) \tag{24}$$

식 (22)-(24)에서의 m_n 은 $m_n = [t_n / T_s]$ 를 만족하는 양의 정수이며, $[\cdot]$ 는 가우스 기호를 나타낸다. 위의 나눗셈으로 인해 발생하는 나머지로 인한 오차에 대한 보상 방법은 참고문헌 [14]에 제시되어 있으며, 본 논문에서는 샘플링 시간이 충분이 작다는 가정하에 나머지로 인하여 생기는 오차는 무시하도록 한다.

V. 궤적 생성기 구성도

III 장에서 설명한 컨볼루션을 이용하여 구성된 궤적 생성기를 그림 7에 나타내었다. 그림 7에서 볼 수 있듯이, 제안된 궤적 생성기는 매개 변수 계산기(parameter calculator)와 컨볼루션 연산기(convolution operator)로 구성된다. 매개 변수 계산

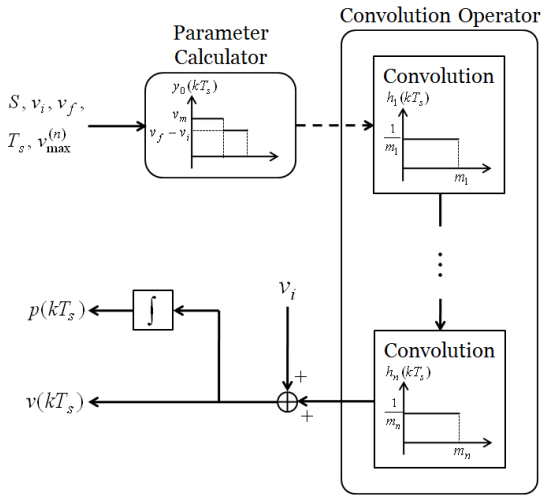

그림 7. 궤적 생성기의 구성도.

Fig. 7. Block diagram of trajectory generator.

기에서는 이동 거리, 초기 속도, 최종 속도, 샘플링 시간, 그리고 시스템 상수인 최대 속도, 최대 가속도 등을 입력으로 받아 v_0, m_n, t_n 을 출력으로 내어 놓는다. 컨볼루션 연산기에서는 매 샘플링 시간마다 n 번의 컨볼루션 연산을 진행하며 n 차 속도함수인 $v^{(n)}$ 까지 특정 값으로 제한된 궤적을 생성한다. IV 장에서 언급했듯이, 한 번의 컨볼루션은 두 번의 덧셈과 한번의 나눗셈으로 연산이 가능하며, n 번의 컨볼루션 연산이 진행될 경우 매 샘플링 시간마다 총 $2*n$ 번의 덧셈과 n 번의 나눗셈을 수행하는데 필요한 연산 시간이 소요된다. 그림 5에서와 같이 실제 결과 값은 컨볼루션을 통해 연산된 결과와 초기속도 v_i 와의 합성을 통해 나타내어지므로 컨볼 루션 연산기에서 출력된 결과는 초기 속도 v_i 와 더해져 매 샘플링 시간에서의 속도 $v(kT_s)$ 를 출력한다. 다항식을 이용한 궤적 생성의 경우, 초기 및 최종 속도 설정을 위해서는 3차 이상의 다항식을 이용하여야 한다.

$$p(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{25}$$

식 (25)는 속도 설정을 위한 위치 영역에서의 3차 다항식이며, 식에서 볼 수 있듯이 매 시간 총 세 번의 덧셈과 다섯 번의 곱셈이 필요하다. 반면, 본 논문에서 제안하는 궤적 생성 방법의 경우, 저크를 바운드 시키기 위해서는 두 번의 컨볼루션이 필요하며, 위치 결과를 출력하기까지 한 번의 덧셈이 추가되어 총 다섯 번의 덧셈과 두 번의 곱셈이 필요하다. CISC 기반 프로세서 기준으로, 컨볼루션을 이용한 궤적 생성 방법은 다항식을 이용한 궤적 생성과 비교하였을 때 비슷한 계산 양이 요구된다는 것을 알 수 있다.

그림 8은 컨볼루션 연산기 내 하나의 컨볼루션 연산의 구성도를 나타낸 것이다. 디지털 컨볼루션은 결과 값의 한 단계 전의 결과가 현재의 연산에 영향을 미치며, 또한 입력 값의 한 단계 전의 값이 현재의 연산에 영향을 미친다. 따라서 궤적 생성기는 전 과정의 결과 값 저장을 위한 한 개의 저장 공간과 m_n 과정 전의 입력 값 저장을 위한 m_n 개의 저장 공간 즉, 총 $(m_n + 1)$ 개의 저장 공간을 필요로 한다.

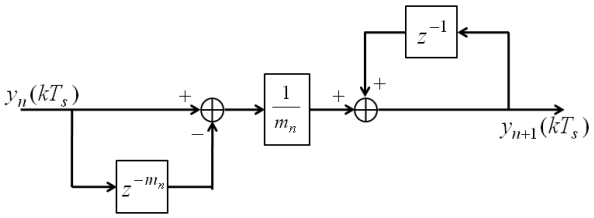

그림 8. 컨볼루션 연산기의 구성도.
Fig. 8. Block diagram of convolution operator.

VI. 모의 실험

제안된 궤적 생성 방법을 MATLAB을 이용하여 모의 실험을 수행하였으며, V 장에서 설명한 구성도를 바탕으로 알고리즘을 구현하였다. 모의 실험은 초기 및 최종 속도 조건이 '0'일 때와 '0'이 아닐 때에 대하여 각각 수행하였다. 모의 실험을 위한 입력 값 S, v_i, v_f, $v_{\max}$, $a_{\max}$, $j_{\max}$, T_s 들은 임의로 설정하였으며, 실험의 편의를 위해 단축상의 점과 점 사이의 운동을 통하여 실험 결과를 보였다.

1. '0'인 속도 조건

표 1과 같은 임의의 입력에 대하여 실험을 수행하였다. 실험 1의 경우, 그림 9에 보여진 결과와 같으며 각각의 속도, 가속도 및 저크는 주어진 n 차 속도함수의 최대값에 도달하며 그 값에 바운드 됨을 확인할 수 있다. 반면, 실험 2의 결과인 그림 10를 보면, n 차 속도함수가 주어진 값에 바운드 되어있으나 속도의 경우, 주어진 값에 도달을 하지 못하는 것을 확인 할 수 있다. 표 2는 매개 변수 계산기의 출력으로 나오는 변수들의 결과값이다. 실험 1의 경우 식 (6)의 조건을 만족하나, 실험 2의 경우는 그렇지 못하다. 이는 주어진 저크와 가속도로 최대 속도까지 변속을 하기 위해 필요한 시간이 충분히 주어지지 않는다는 의미이기도 하다. 따라서, III 장에서 언급했듯이 n 차 속도함수 $v^{(n)}$ 가 최대 값에 도달하기 위해서는 식 (6)의 조건을 만족하여야 한다.

2. '0'이 아닌 속도 조건

'0'이 아닌 속도 조건에 대해서 표 3과 같은 임의의 입력에 대하여 실험을 수행하였다. 그림 11과 그림 12는 실험 3과 실험 4에 대한 각각의 결과 그래프이다. 각각의 경우 주어진 제한 조건과 주어진 초기 및 최종 속도를 만족하는 궤

표 1. '0'인 속도 조건에서의 모의 실험의 입력 값.
Table 1. Input parameters for simulation of zero-velocity condition.

	실험1	실험2		실험1	실험2
$S[m]$	8	8	$v_{\max}[m/s]$	4	4
$v_i[m/s]$	0	0	$a_{\max}[m/s^2]$	4	2
$v_f[m/s]$	0	0	$j_{\max}[m/s^3]$	8	2
$T_s[s]$	0.01	0.01			

표 2. 매개 변수 계산기의 출력 값.
Table 2. Output parameters in parameter calculator.

	$v_m[m/s]$	$t_x[s]$	$t_1[s]$	$t_2[s]$
실험1	4	2	1	0.5
실험2	4	2	2	1

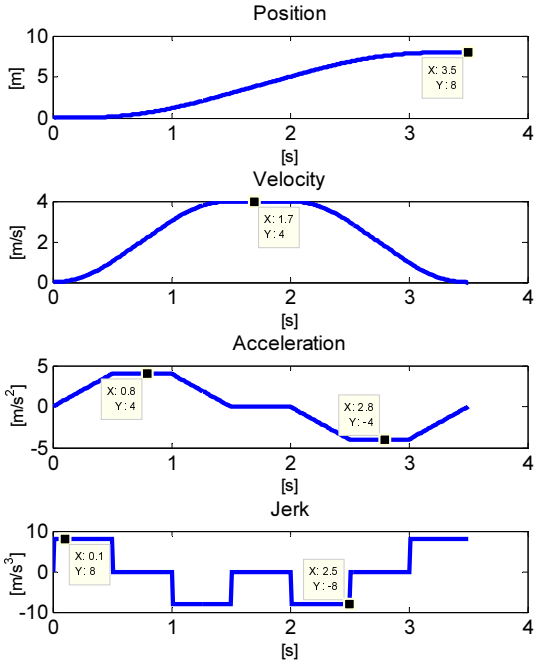

그림 9. 실험1의 결과 그래프.
Fig. 9. Simulation result #1.

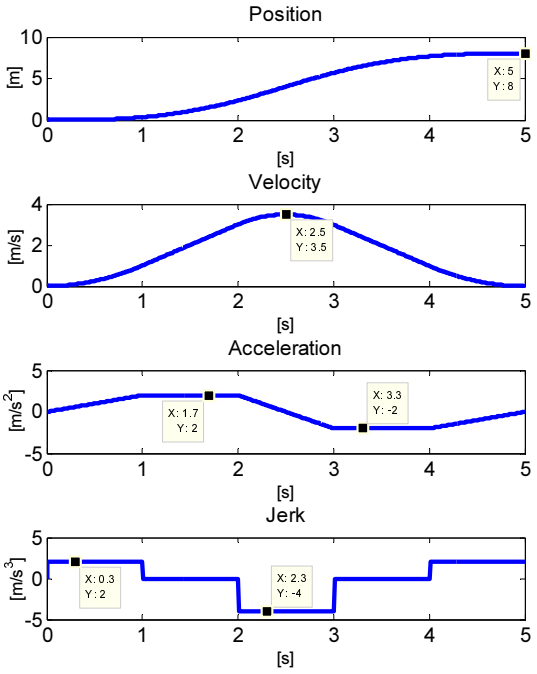

그림 10. 실험2의 결과 그래프.
Fig. 10. Simulation result #2.

표 3. '0'이 아닌 속도 조건에서의 모의 실험의 입력 값.
Table 3. Input parameters for simulation of nonzero-velocity condition.

	실험3	실험4		실험3	실험4
$S[m]$	3	5	$v_{\max}[m/s]$	4	4
$v_i[m/s]$	1	-2	$a_{\max}[m/s^2]$	1	8
$v_f[m/s]$	-3	1	$j_{\max}[m/s^3]$	4	16
$T_s[s]$	0.01	0.01			

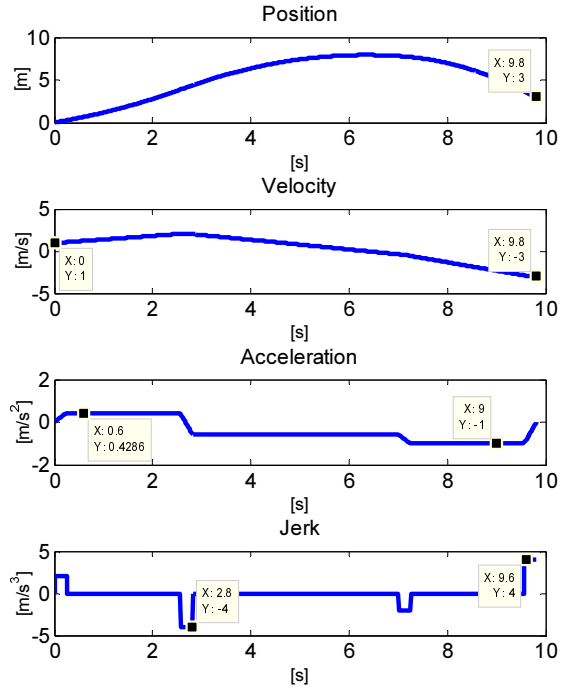

그림 11. 실험3의 결과 그래프.
Fig. 11. Simulation result #3.

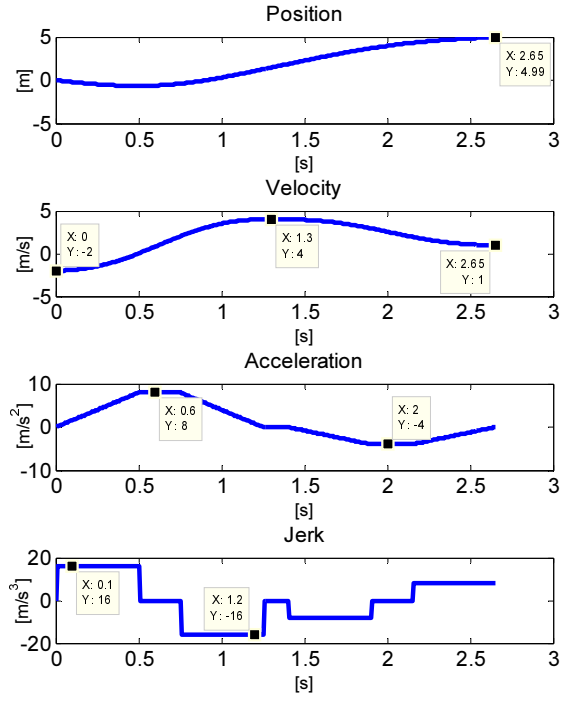

그림 12. 실험4의 결과 그래프.
Fig. 12. Simulation result #4.

적이 생성되는 것을 볼 수 있다. 또한 식 (6)의 조건에 따라 생성된 궤적이 최고 속도에 도달하는 경우와, 그렇지 못하는 경우를 볼 수 있다. 또한, 그림 8과 그림 12을 보면 특성 4에서와 같이, 변속 후 그 속도가 일정 시간 유지 되는 경우 속도 함수의 가속도 및 감속도 특성이 대칭적으로 형성 되는 것을 확인 할 수 있다.

VII. 결론

본 논문에서는 디지털 컨볼루션을 이용한 시스템 제한이 적용된 궤적 생성 방법을 제안하였다. 제안된 궤적 생성 방법은 속도, 가속도 그리고 저크와 같은 n 차 속도 상수를 원하는 값에 제한시킴으로 제어 시스템에 발생될 수 있는 원치 않은 손상을 줄일 수 있다. 또한, 시스템의 최대 속도와 가속도를 궤적 생성시 반영하기 때문에, 즉 시스템의 최대 사양을 반영 한 궤적을 생성하기 때문에 제어기를 통한 전체 제어시스템의 더 정확하고 빠른 추정을 가능하게 한다. 또한 디지털 컨볼루션의 방법은 다항식을 이용한 방법과 비슷한 계산 량으로 처리 속도가 빠르지 않은 작은 칩 기반의 시스템에서도 실시간으로의 동작 구현이 가능하다. 마지막으로 제안된 방법을 단축상의 점과 점 사이의 운동에 관해 MATLAB을 이용한 모의 실험을 통하여 타당성을 검증하였으며, 입력 값에 따른 생성 곡선의 특성을 보여주었다.

로봇의 움직임에는 힘과 관련된 동작이 상당한 비중을 차지하며, 힘을 제어하기 위해서는 가속도 설정이 가능한 궤적이 필요하다. 또한, 제안된 방법은 컨볼루션 연산의 특성상 속도 증가 구간과 속도 감소 구간의 시간이 같기 때문에 각 구간에서의 최대 가속도는 다르게 나타난다. 이에 컨볼루션의 변형을 통해 가속도 설정이 가능한 궤적 생성, 또는 두 구간의 최대 가속도가 같은 궤적 생성이 가능하다면 좀 더 효율적인 궤적이 될 수 있을 것이라 기대한다.

참고문헌

[1] J. J. Craig, *Introduction to Robotics*, New York: Addison-Wesley, 1989.

[2] D. Constantinescu and E. A. Croft, “Smooth and time-optimal trajectory planning for industrial manipulators along specified paths,” *Trans. on Journal of Robotic System*, vol. 17, pp. 233-249, 2000.

[3] Y. Mizoshita, S. Hasegawa, and K. Takaishi, “Vibration minimized access control for disk drives,” *IEEE Trans. on Magnetics*, vol. 32, pp. 1793-1798, May 1996.

[4] J. W. Jeon and Y. Y. Ha, “A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools,” *IEEE Trans. on Industrial Electronics*, vol. 47, pp. 133-139, Feb. 2000.

[5] K. T. Ahn, J. S. Cho, and W. K. Chung, “Discrete trajectory formation in comparison with the analytical method for smooth movements,” *Proc. of IEEE Int. conf. on Industrial Electronics*, Paris, pp. 4462-4467, Nov. 2006.

[6] S. Macfarlane and E. A. Croft, “Jerk-bounded manipulator trajectory planning: design for real-time applications,” *IEEE Trans. on Robotics and Automation,* vol. 19, pp. 42-52, Feb. 2003.

[7] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision and Intelligence*. New York, McGraw-Hill, 1987.

[8] S. -H. Nam and M.-Y. Yang, “A study on a generalized parametric interpolator with real-time jerk-limited acceleration,” *Trans. on Computer-Aided Design,* vol. 36, pp. 27-36, Jan. 2004.

[9] K. D. Nguyen, I. -M. Chen, and T. -C. Ng, “Planning algorithms for s-curve trajectories,” *Proc. of Int. Conf. on Advanced Intelligent Mechatronics,* pp. 1-6, Sep. 2007.

[10] P. Lambreshts, M. Boerlage, and M. Steinbuch, “Trajectory

planning and feedforward design for electromechanical motion systems," *Trans. on Control Engineering Practice*, vol. 13, pp. 145-157, Feb. 2005.

[11] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation," *Trans. on Int. Journal of Machine Tools and Manufacture*, vol. 41, pp. 1323-1345, Jul. 2001.

[12] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Trans. on Int. Journal of Robotics Research*, vol. 4, pp. 3-17, 1985.

[13] D. S. Khalsa, "High performance motion control trajectory commands based on the convolution integral and digital filtering," *Proc. of Int. Conf. on Intelligent Motion*, pp. 54-61, Oct. 1990.

[14] K. -H. Su, S. -K. Hu, and M. -Y. Cheng, "Design and implementation of an FPGA-based motion command generation chip," *Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics,* pp. 5030-5035, Oct. 2006.

이 건

2008년 한양대학교 전자컴퓨터공학부(공학사). 2008년~현재 동 대학 전자전기제어계측공학과 석사과정 재학. 관심분야는 궤적생성, 장애물회피, 견실제어.

김 진 현

1998년 포항공과대학교 기계공학과(공학사). 2000년 동 대학원(공학석사). 2005년 동 대학원(공학박사). 2005년 한국생산기술연구원 로봇기술본부 선임연구원. 2007년~현재 서울과학기술대학교 기계공학과 조교수. 관심분야는 여유자유도로봇, 수중로봇, 의료로봇.

최 영 진

1994년 한양대학교 정밀기계공학과(공학사). 1996년 포항공과대학교 기계공학과(공학석사). 2002년 동 대학원(공학박사). 2002년~2005년 한국과학기술연구원(KIST) 지능로봇연구센터 선임연구원. 2005년~현재 한양대학교 전자시스템공학과 조교수. 관심분야는 Bio-Robotics, 로봇제어이론, 재활로봇, PID 제어이론.