2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

New Robotic Motion Generation using Digital Convolution with Physical System Limitation

Geon Lee, Byung-Ju Yi, Doik Kim and Youngjin Choi, *Member, IEEE*

Abstract—**This paper proposes a novel trajectory generation method considering physical system limitations such as maximum velocity, acceleration, and jerk. Also, the trajectory generation method uses a digital convolution for reducing computational load in robotic motion applications. The method to be proposed has several advantages; first, a continuously differentiable trajectory is simply obtained by applying successive convolution operations. Second, a resultant trajectory is always generated satisfying the given physical system limits. Third, the suggested method has low computational load thanks to recursive form of convolution operation. The effectiveness of the proposed method is shown through numerical simulations.**

I. Introduction

THE most important thing in the trajectory generation is to make the position trajectory of S-curve shape function which is differentiable at least till either acceleration-level or jerk-level. Especially, a jerk trajectory bounded within the physical jerk limit of the actuator specification will reduce damages to the control system from unforeseen vibrations or overshoots as well as improve the accuracy or speed when the system is under tracking [1]. Also, the smaller the jerk is limited, the smoother the trajectory is generated [2]. For a special purpose such as surgical robot that makes contact with patients, the jerk-level trajectory bounded with an arbitrary value must be considered rather than just the bounded acceleration-level trajectory. Moreover, the jerk-level is required to be kept as small as possible [3, 4].

The actuator has its own physical limits such as maximum velocity, acceleration, and jerk. To realize the control system with higher performance, the physical system limits should be considered when the desired trajectory is generated [2,5,6]. The trajectory generated over the physical system limits is not only impossible to be followed by the controller, but also gives damages to the system due to overload of the actuator. If we make use of the trajectory generator without considering the system limits, we may spend a lot of time to find experimentally a range of the available trajectory. Last but not least, the economic cost for realizing the trajectory generator is an important element for extending application ranges to industrial and household control systems. If the control system including the trajectory generator can be implemented in a cheap processor, it is able to reduce the economic cost in realizing a real-time control system [7, 8].

Generally, the desired jerk trajectory has been generated using higher order polynomial method above third order [9]. The polynomial method is very useful because it can establish individual functions of velocity, acceleration, jerk, and so forth by changing its order. However, the conventional polynomial method has a disadvantage that it cannot satisfy the given physical system limits. As an alternative, several methods have been suggested to satisfy the physical system limits by using the polynomial method. The most intuitive method makes the desired trajectory divide into many segments. For example, the trajectory generation including arbitrary jerk limit requires sixteen segments using forth order polynomial functions [12]. Also, many other methods to remedy this disadvantage have been proposed with the purpose of lower computational burden in [11-14]. On the other hand, a convolution-based trajectory generation method has been suggested in [5, 15], which does not use the polynomial any more. The convolution-based method is able to generate an S-curve trajectory by applying successive convolution operations. Also a recursive form of convolution operation can reduce the computational loads drastically. However, the conventional convolution-based trajectory generation method has been developed under only zero initial and terminal conditions. In this paper, we are to extend it to more general case including non-zero initial and terminal conditions as well as to establish it on the theoretically substantial basis.

II. Trajectory Generation Method: Zero States

Before suggesting a novel convolution-based trajectory generation method, this section reviews several properties of convolution operations. Here we assume that the impulse response of the system, $h(t)$, is a rectangular function having a unit area. Then the output can be obtained as a form of smoother function than the input by applying the input to the system, only if the input is a piecewise continuous function. Using this property, it is possible to generate the smooth desired trajectory by applying successive convolution operations.

Suppose that $x(t)$ is an arbitrary input function defined in time duration of $0 \le t \le t_x$, $h(t)$ is a convoluted rectangular function having the unit area defined in time duration of $0 \le t \le t_h$, namely, $h(t) = 1/t_h$, for $0 \le t \le t_h$, and $y(t)$ is an output function produced by the convolution operation on

This work was supported in part by the Mid-career Researcher Program through NRF grant funded by the MEST (No. 2008-0061778), and in part

G. Lee, B-J. Yi and Y. Choi are with the Department of Electronics, Electrical, Control and Instrumentation Engineering, Hanyang University, South Korea (e-mail: {emerald,bj,cyj}@hanyang.ac.kr).

D. Kim is with the Korea Institute of Science and Technology, South Korea (e-mail: doikkim@kist.re.kr)

two functions $x(t)$ and $h(t)$. Here, we should note that two functions $x(t)$ and $h(t)$ are zeros outside the defined time durations. Also let us denote that x_m and y_m are the maximum values of $x(t)$ and $y(t)$, respectively, then the convolution operations have several properties as follows;

Property 1 The output $y(t)$ is defined in time duration of $0 \le t \le t_x + t_h$, which is the sum of both time durations of the input $x(t)$ and the convoluted function $h(t)$.

Property 2 The area of output $y(t)$ is always equal to that of input $x(t)$.

Property 3 The maximum absolute value of output y_m is always smaller than or equal to that of input x_m. Especially, if $x(t)$ maintains x_m constantly for the time duration t_h or more, then y_m is equal to x_m.

First, it is easy to prove the Property 1 using the formal definition of convolution operation as follows:

$$
\begin{aligned}
y(t) &= \int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \\
&= \frac{1}{t_h}\int_0^{t_h} x(t-\tau)d\tau \\
&= \begin{cases} \dfrac{1}{t_h}\displaystyle\int_0^{t_h} x(t-\tau)d\tau, & 0 \le t \le t_x + t_h \\ 0, & otherwise \end{cases}
\end{aligned}
\tag{1}
$$

This property comes out from convolution operation itself. Extending this property to successive convolution applications, we can know that the result function is defined in the total sum of time durations of the input and the convoluted functions.

Second, suppose that $X(s)$, $H(s)$, and $Y(s)$ are the Laplace transforms of $x(t)$, $h(t)$, and $y(t)$, respectively, then the area of $y(t)$ is denoted by $Y(s)/s$ and the convolution operation on two functions $x(t)$ and $h(t)$ implies the multiplication of $X(s)$ and $H(s)$. Now, the Property 2 is proved easily by using the final value theorem as follows:

$$
\begin{aligned}
\lim_{s\to 0} s\left(\frac{Y(s)}{s}\right) &= \lim_{s\to 0} X(s)H(s) \\
&= \lim_{s\to 0}\left(X(s)\frac{1}{t_h s}\left(1-e^{-t_h s}\right)\right) \\
&= \lim_{s\to 0} X(s)
\end{aligned}
\tag{2}
$$

where we should note that the Laplace transform of $h(t) = 1/t_h$ for $0 \le t \le t_h$ is $H(s) = (1-e^{-t_h s})/(t_h s)$ and l'Hôpital's rule was used in Eq. (2). Above equation implies that the area of input is always equal to that of output. This completes the proof of the Property 2. Moreover, the invariant area principle is always true only if the convoluted functions have the unit area.

Third, let us assume that $y(t)$ has the maximum value y_m at any time t_m, then we can get the following relation from the Eq. (1):

$$
\begin{aligned}
y(t_m) = y_m &= \frac{1}{t_h}\int_0^{t_h} x(t_m-\tau)d\tau \\
&\le \frac{1}{t_h}\int_0^{t_h} x_m d\tau = x_m
\end{aligned}
\tag{3}
$$

where we can know that above inequality is always true because we chose the maximum value x_m among all values of function $x(t)$. Namely, the maximum value of the output is smaller than or equal to that of the input. Also, if $x(t) = x_m$ for $0 < t_h \le t_x$, then $y_m = x_m$. These complete the proof of the Property 3. In other words, the maximum value of the output cannot exceed that of the input only if the convoluted functions have the unit area.

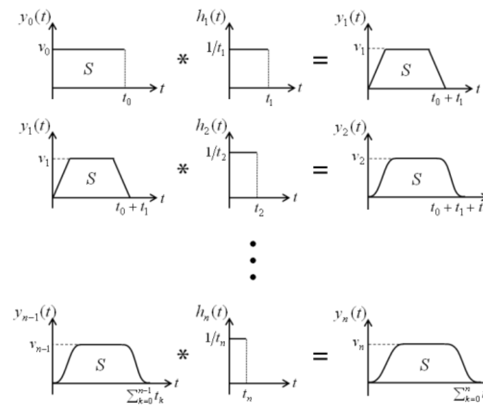

Fig. 1. Convolution-based trajectory generation: zero states.

A. Zero-initial and Zero-terminal Conditions

For the sake of simplicity, we will consider a single-axis motion control system as a target system. Also, suppose that the system has the limits such as the maximum velocity denoted by $v_{\max}$, the maximum acceleration denoted by $v_{\max}^{(1)}$, and the maximum jerk denoted by $v_{\max}^{(2)}$. Without loss of generality, the system limit for *n*th order differentiation of velocity function is denoted by $v_{\max}^{(n)}$. Also, assume that the system moves given distance, S, then we can make the input function $y_0(t)$ using the maximum velocity $v_{\max}$ as follows:

$$
y_0(t) = \begin{cases} v_0 \ , & 0 \le t \le t_0 \\ 0 \ , & otherwise \end{cases}
\tag{4.a}
$$

$$
\mathrm{v}_0 = \mathrm{sgn(S)v_{max}} \quad \text{and} \quad \mathrm{t}_0 = \frac{|\mathrm{S}|}{\mathrm{v_{max}}}
\tag{4.b}
$$

where v_o and t_0 imply the signed maximum velocity and time duration of the rectangular input function as shown in Fig. 1.

Now let us define the first convoluted function $h_1(t)$ as the rectangular function with the time duration $0 \le t \le t_1$, then we can get the trapezoidal function $y_1(t)$ produced by the convolution operation on two rectangular functions $y_0(t)$ and $h_1(t)$ as shown in Fig. 1. By Property 1, the output function $y_1(t)$ is defined in the time duration $0 \le t \le t_0 + t_1$. By the Property 2, the distance to be moved S is not changed by the convolution. By the Property 3, the maximum absolute value $|v_1|$ of $y_1(t)$ becomes smaller than or equal to the maximum velocity $v_{\max}$ of the given system. From the trapezoidal function $y_1(t)$ of the Fig. 1, we can know that each time duration for acceleration and deceleration is t_1, respectively, thus the maximum acceleration of $y_1(t)$ becomes $v_{\max}/t_1$. To make active use of the physical limit about maximum acceleration $v_{\max}^{(1)}$, let us determine the time duration of first convoluted function $h_1(t)$ to be $t_1 = v_{\max}/v_{\max}^{(1)}$. Moreover, if $y_0(t)$ maintains the signed maximum v_0 constantly for the time interval t_1 or more, namely, if $t_0 \ge t_1$ in the Fig. 1, then $v_1 = v_0$ by the Property 3.

Let us apply the convolution operation once more using the trapezoidal velocity function $y_1(t)$ as the input. Similarly we define the second convoluted function $h_2(t)$ as the rectangular function with time duration $0 \le t \le t_2$, then we can get the S-curve function $y_2(t)$ produced by the convolution of $y_1(t)$ and $h_2(t)$ as shown in Fig. 1. By Property 1, the output function $y_2(t)$ is defined in time duration $0 \le t \le t_0 + t_1 + t_2$. By the Property 2, the distance S is also not changed through the convolution. By the Property 3, the maximum absolute value $|v_2|$ of $y_2(t)$ becomes smaller than or equal to the maximum velocity $v_{\max}$ of the given system. To make active use of the physical limit about maximum jerk, we determine the time duration to be $t_2 = v_{\max}^{(1)}/v_{\max}^{(2)}$. Also, if $y_1(t)$ maintains the maximum v_1 constantly for the time duration t_2 or more, namely, if $t_0 - t_1 \ge t_2$ and $t_1 \ge t_2$ in $y_1(t)$ of the Fig. 1, then $v_2 = v_1$ by the Property 3.

Without loss of generality, we can extend above procedures to the smoother S-curve velocity function within the allowable physical system limits such as $v_{\max}^{(n)}, \cdots, v_{\max}^{(2)}, v_{\max}^{(1)}$ and $v_{\max}$. Here, only design parameters to be considered are the time durations of the convoluted functions, which should be determined as follows:

$$t_k = \frac{v_{\max}^{(k-1)}}{v_{\max}^{(k)}} \quad for \;\; k = 0, 1, 2, \cdots, n \tag{5}$$

where $v_{\max}^{(0)} = v_{\max}$ and $v_{\max}^{(-1)} = |S|$. Also, in order to generate the efficient trajectory by making active use of all the physical system limits such as $v_{\max}^{(n)}, \cdots, v_{\max}^{(2)}, v_{\max}^{(1)}$ and $v_{\max}$, the following inequality conditions as regards the time durations should be satisfied:

$$t_l \ge \sum_{k=l+1}^{n} t_k \qquad for \quad l = 0, 1, 2, \cdots, n \tag{6}$$

In summary, for given the distance S and the physical system limits $v_{\max}^{(n)}, \cdots, v_{\max}^{(2)}, v_{\max}^{(1)}, v_{\max}$, if we determine the time durations by using Eq. (5), then the S-curve velocity function generated by the suggested convolution-based method is always within the allowable physical limits or at their boundary. As mentioned before, the convolution-based trajectory generation method is effective because it satisfies automatically the physical system limits, but the continuous-time convolution operations require much computational burden. As an alternative, a recursive form of convolution operation to reduce it drastically will be proposed later.

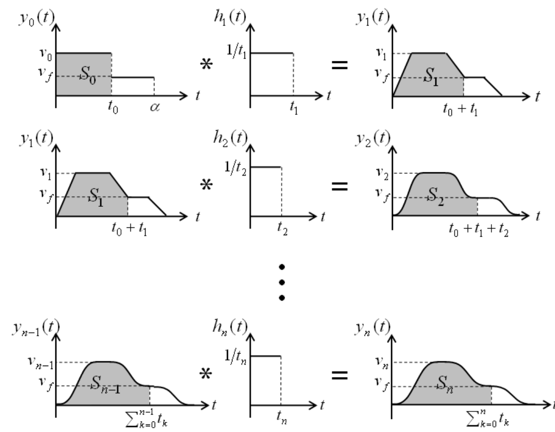

Fig. 2. Convolution-based trajectory generation method: non-zero terminal condition.

III. TRAJECTORY GENERATION: NON-ZERO STATES

A. Non-zero Terminal Condition

In order to generate desired trajectory with non-zero terminal velocity, let us consider the input of stepwise function that is composed of the maximum velocity v_0 for $0 \le t \le t_0$ and terminal velocity v_f for $t_0 \le t \le \alpha$, where α is a sufficiently large time parameter. Actually, the value of α is dependent on the number of convolution operations to be applied, which is expressed by $\alpha = \sum_{k=0}^{n} t_k$; e.g., if *n=2*, then $\alpha = t_0 + t_1 + t_2$ and if *n=3*, then $\alpha = t_0 + t_1 + t_2 + t_3$. If the stepwise input $y_0(t)$ is convoluted with the unit area function $h_1(t)$, then we have the output $y_1(t)$ as shown in Fig. 2. Once more, if we perform the convolution operation on two functions of $y_1(t)$ and $h_2(t)$, then the S-curve function is generated as shown in Fig. 2. Here, by the Property 3, the maximum velocity of output function v_n is smaller than or equal to that of the stepwise input v_0. Also, the output $y_n(t)$ goes to zero passing through v_f at the time $t=\sum_{k=0}^{n} t_k$ as shown in Fig. 2. Hence, the trajectory satisfying the non-zero terminal condition can be generated by performing the successive convolution operations until $t=\sum_{k=0}^{n} t_k$. As we can see in the Fig. 2, the convolution time duration in the case of non-zero terminal condition is equal to that of zero states suggested in the previous section. Namely, the time duration of the convolution-based trajectory generation method is independent of the value of terminal velocity.

In the case of non-zero terminal condition, the distance to be moved becomes different according to the number of convolutions. Since the acceleration and deceleration during time interval of $t_0 \le t \le \sum_{k=0}^{n} t_k$ always are formed symmetrically regardless of the number of successive convolution operations, the area S_n of final velocity function $y_n(t)$ with the non-zero terminal condition is obtained as follow:

$$\begin{aligned} S_n &= S_0 + v_f(\alpha - t_0) - \frac{v_f}{2}\sum_{k=1}^{n} t_k \\ &= v_0 t_0 + v_f\left(\sum_{k=0}^{n} t_k - t_0\right) - \frac{v_f}{2}\sum_{k=1}^{n} t_k \\ &= v_0 t_0 + \frac{v_f}{2}\sum_{k=1}^{n} t_k \end{aligned} \tag{7}$$

where we can see that the Property 2 holds if the terminal velocity condition is zero.

B. Non-zero Initial Condition

If the trajectory should have both non-zero initial and terminal velocity conditions, then the trajectory can be decomposed into a rectangular initial velocity function and the non-zero final velocity function as shown in Fig. 3. The rectangular function has a value of initial velocity v_i for $0 \le t \le \sum_{k=0}^{n} t_k$ and the non-zero final velocity function has a difference between terminal and initial conditions $v_f - v_i$ at the terminal time as shown in Fig. 3. Also, its area S_n in the case of non-zero initial and terminal conditions can be obtained as the sum of areas of two functions, S_n^t and S_n^b , as following form:

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

where we can see that the Property 2 holds if both initial and terminal conditions are zero. In the case of non-zero initial and terminal conditions, the distance to be moved becomes different according to the number of convolutions. Since the area S_n of the final velocity function $y_n(t)$ should be equal to the given distance S of the system, we are to modify how to determine v_0 differently from Eq. (4.b). Accompanied by the change of v_0, other two parameters t_0 and t_1 should be also modified. The following part suggests how these three parameters (v_0, t_0, t_1) should be changed in order to satisfy the given distance, namely, $S_n = S$.

For given v_i and v_f , one of four possible trajectories can

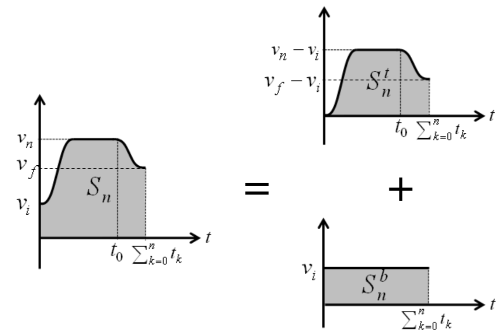

Fig. 3. Decomposition of the trajectory with non-zero states into a rectangular initial velocity function and the non-zero final velocity function.

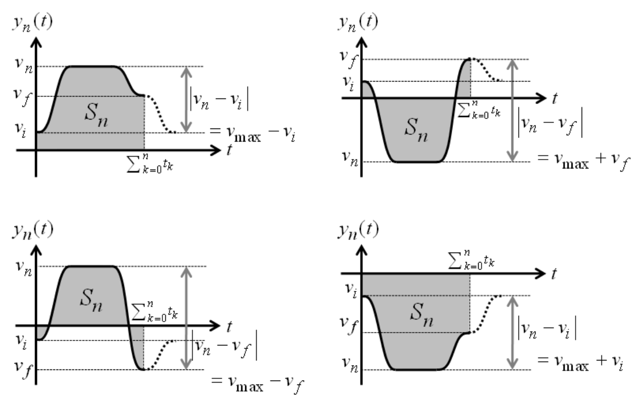

Fig. 4. Four possible trajectories according to the given distance, initial and terminal conditions.

TABLE I

THE VALUE OF t_1^* ACCORDING TO RELATIONSHIP BETWEEN v_i AND v_f .

	$v_i \ge 0$			$v_i < 0$	
	$\lvert v_i \rvert \ge \lvert v_f \rvert$	$\lvert v_i \rvert < \lvert v_f \rvert$		$\lvert v_i \rvert \ge \lvert v_f \rvert$	$\lvert v_i \rvert < \lvert v_f \rvert$
$v_f \ge 0$	$\frac{v_{\max} - v_f}{v_{\max}^{(1)}}$	$\frac{v_{\max} - v_i}{v_{\max}^{(1)}}$	$v_f \ge 0$	$\frac{v_{\max} + v_f}{v_{\max}^{(1)}}$	$\frac{v_{\max} - v_i}{v_{\max}^{(1)}}$
$v_f < 0$	$\frac{v_{\max} - v_f}{v_{\max}^{(1)}}$	$\frac{v_{\max} + v_i}{v_{\max}^{(1)}}$	$v_f < 0$	$\frac{v_{\max} + v_f}{v_{\max}^{(1)}}$	$\frac{v_{\max} + v_i}{v_{\max}^{(1)}}$

generally be generated according to the distance to be moved S_n as shown in Fig. 4, upper two trajectories in the Fig. 4 are for the case of $v_i < v_f$ and below ones for the case the of $v_i > v_f$. Also, the maximum velocity of the generated trajectory can be either $v_n = v_{\max}$ or $v_n = -v_{\max}$ according to the given distance. Now, let us introduce a concept of criterion distance denoted by S_n^* , namely, two trajectories shown in left ones of the Fig. 4 are generated using $v_n = v_{\max}$ when $S_n > S_n^*$ and two trajectories shown in right ones of the Fig. 4 are generated using $v_n = -v_{\max}$ when $S_n < S_n^*$. In order to find out the criterion distance S_n^* , if we approach t_0 to zero in the Fig. 2, then we have Fig. 5. The Fig. 5 shows an inevitable distance for moving from zero to terminal velocity under the case of zero initial velocity. In the case of non-zero initial velocity, corresponding decomposition like the Fig. 3 should be considered. Moreover, since the S_n^* is dependent on the value of t_1 , we take the value of S_n^* from the minimum t_1^* determined as a small value between $|v_n - v_i| / v_{\max}^{(1)}$ and $|v_n - v_f| / v_{\max}^{(1)}$. More detail, since v_n can be either $v_{\max}$ or $-v_{\max}$, we have possible eight cases of the minimum t_1^* according to the relationship between v_i and v_f as shown in Table I. Fortunately, the possible eight cases can be expressed by one equation as the following form:

$$t_1^* = \frac{v_{\max} - \operatorname{sgn}(v_i v_f)\min(|v_i|, |v_f|)}{v_{\max}^{(1)}} \tag{9}$$

Once the value of t_1^* is determined, the criterion distance can be obtained from Eq. (8) by $t_0 \to 0$ as follow:

$$S_n^* = \frac{v_f + v_i}{2}\left(t_1^* + \sum_{k=2}^{n} t_k\right) \tag{10}$$

Since the generated final velocity function should be bounded as the maximum velocity of the given system, the value of v_0 can be either $v_{\max} - v_i$ when $S_n > S_n^*$ or $-v_{\max} - v_i$ when $S_n < S_n^*$. Also, the value of v_0 can take any value when $S_n = S_n^*$ because t_0 will be zero in that case. Thus, the value of v_0 can be expressed by either

$$v_0 = \operatorname{sgn}(S_n - S_n^*) v_{\max} - v_i \text{ or } v_0 = \frac{1}{\operatorname{sgn}(S_n - S_n^*)} v_{\max} - v_i \tag{11}$$

If $S_n > S_n^*$, then we can see that the value of t_1 should take the large value between $(v_{\max} - v_f)/v_{\max}^{(1)}$ and $(v_{\max} - v_i)/v_{\max}^{(1)}$ from left two trajectories in the Fig. 4, on the other hand, if $S_n < S_n^*$, then the value of t_1 should take the large value between $(v_{\max} + v_f)/v_{\max}^{(1)}$ and $(v_{\max} + v_i)/v_{\max}^{(1)}$ from right two trajectories in the Fig. 4. Thus, the value of t_1 is expressed as following form:

$$t_1 = \frac{\max\left(v_{\max} - \operatorname{sgn}(S_n - S_n^*) v_i,\; v_{\max} - \operatorname{sgn}(S_n - S_n^*) v_f\right)}{\frac{1}{2}\left(1 + \operatorname{sgn}(S_n - S_n^*)^2\right) v_{\max}^{(1)}} \tag{12}$$

where we should note that we take $t_1 = 2v_{\max} / v_{\max}^{(1)}$ when $S_n = S_n^*$. Finally, the value of t_0 can be obtained by applying Eq. (11) to Eq. (8) as follow:

$$t_0 = \frac{\operatorname{sgn}(S_n - S_n^*)}{v_{\max}}\left(S_n - \frac{v_f + v_i}{2}\sum_{k=1}^{n} t_k\right) \tag{13}$$

Till now, we have suggested how these three parameters (t_0, t_1, v_0) should be changed for satisfying $S_n = S$ in the case of non-zero initial and terminal condition.

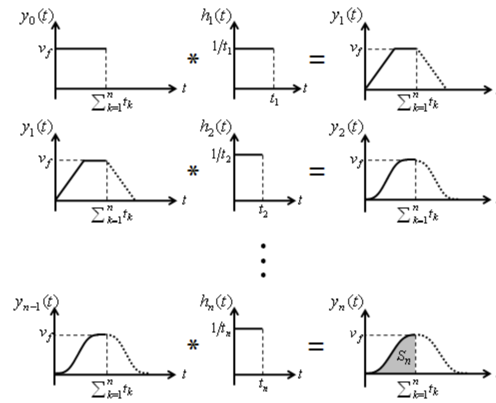

Fig. 5. Criterion distance S_n^* obtained as $t_0 \to 0$ in the Fig. 2, in the case of zero initial velocity.

C. Digital Convolution

The convolution operation can be expressed by two forms such as convolution integral in the continuous time domain and convolution sum in the discrete time domain. Actually, the implementation of convolution integral must be inappropriate for the digital motion control systems, especially for real-time issue. Thus the convolution sum is considered with *n*th convoluted function having unit area, $h_n[k] = 1/m_n$ for $0 \le k \le m_n - 1$, as follows:

$$\begin{aligned} y_n[k] &= \sum_{l=-\infty}^{\infty} h_n[l] y_{n-1}[k-l] \\ &= \frac{1}{m_n}\sum_{l=0}^{m_n-1} y_{n-1}[k-l] \\ &= \frac{1}{m_n}\left(y_{n-1}[k] + y_{n-1}[k-1] + \cdots + y_{n-1}[k-m_n+1]\right) \end{aligned} \tag{14}$$

In addition, $y_n[k-1]$, preceding value of $y_n[k]$, is expressed as following form:

$$y_n[k-1] = \frac{1}{m_n}\left(y_{n-1}[k-1] + y_{n-1}[k-2] + \cdots + y_{n-1}[k-m_n]\right) \tag{15}$$

Subtracting Eq. (15) from Eq. (14), we arrive at a recursive form of the convolution sum as follow:

$$y_n[k] = \frac{y_{n-1}[k] - y_{n-1}[k-m_n]}{m_n} + y_n[k-1] \tag{16}$$

As we can see in the Eq. (16), the recursive form of convolution sum is very effective because it requires just two additions and one division for the convoluted function having unit area. In Eq. (14), (15) and (16), k and m_n are positive integers satisfying $k = [t/T_s]$ and $m_n = [t_n/T_s]$, respectively, with sampling time T_s and Gauss floor function $[x]$ to denote the largest integer not greater than x. An error can be caused by the Gauss floor function according to the size of sampling time in the convolution sum. In this paper, this error will be neglected by adding the assumption that sampling time is enough small. Till now we dealt with several properties of the convolution operation, convolution-based trajectory generation method under the condition of zero states (zero initial and terminal velocities) and non-zero states (non-zero initial and terminal velocities), and the recursive form of convolution sum for real-time implementation issue. The following section will show the effectiveness of the proposed method through simulations

IV. Simulation Results and Concluding Remarks

For the simulation of the proposed method, the input

TABLE II

The input parameters for simulation; as such, system limits, distance to be moved, and initial/terminal conditions.

	$v_{\max}$	$v_{\max}^{(1)}$	$v_{\max}^{(2)}$	v_i	v_f	S
(a)	4[m/s]	4[m/s^2]	16[m/s^3]	2[m/s]	1[m/s]	8[m]
(b)	8[m/s]	4[m/s^2]	16[m/s^3]	2[m/s]	1[m/s]	8[m]

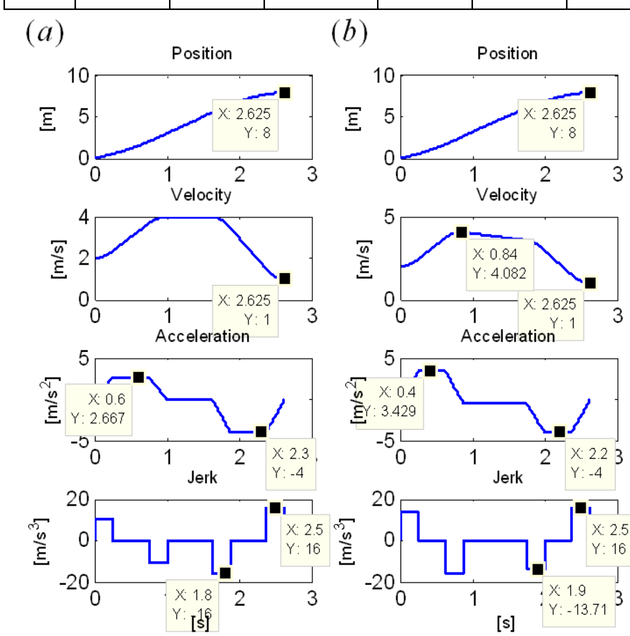

Fig. 6. Simulation results.

parameters are given as shown in Table II. Fig. 6 shows the simulation results. Both results, (a) and (b) in the Fig.6, show that the trajectories are generated within given system limits such as the maximum velocity, the maximum acceleration, and maximum jerk. Comparing (a) with (b), however, the velocity of (b) does not reach to the given maximum velocity, while that of (a) reaches to the given maximum velocity. That is because, in the case of (b), Eq. (6) is not satisfied. We can notice that t_0 takes 0.625 from Eq. (13) and that t_1 and t_2 take 1.75 and 0.25 respectively, so t_0 is smaller than the sum of t_1 and t_2. The proposed method could generate the trajectory under any input parameters, as this result tells us.

The novel trajectory generation method making active use of physical system limits such as maximum velocity, maximum acceleration, and maximum jerk have presented in this paper. The proposed method has utilized the recursive convolution sum, being required two additions and one division per one convolution, for practical use. Through the proposed convolution-based trajectory generation method, we could get a continuously differentiable trajectory simply within the given physical system limits. The suggested method was able to be applicable to both zero and non-zero initial/terminal conditions. Finally, the effectiveness of the suggested method was shown by the numerical simulations.

References

[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," *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] T. Tsuji, Y. Tanaka, P.G. Morasso, V. Sanguincti, and M. Kaneko, "Bio-mimetic trajectory generation of robots via artificial potential field with time base generator," *IEEE Trans. on System, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 32, No. 4, pp. 426-439, 2002.

[5] 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, No.2, pp. 133-139, 2000.

[6] I.M.S. Panahi and H.A. Toliyat, "An efficient processor-based online generation of time-optimal trajectories," *IEEE Trans. on Control Systems Technology*, vol. 14, No. 5, pp. 966-973, 2006.

[7] 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 Control and Instrumentation*, pp. 4462-4467, 2006.

[8] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time application," *IEEE Trans. on Robotics and Automation*, vol. 19, No.1, pp. 42-52, 2003.

[9] M. Schlemmer and S.K. Agrawal, "A computational approach for time-optimal planning of high-rise elevators," *IEEE Trans. on Control Systems Technology*, vol. 10, No. 1, pp. 105-111, 2002.

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

[11] S.-H. Nam and M.-Y. Yang, "A study on a generalized parametric interpolator with real-time jerk-limited acceleration," *Computer-Aided Design*, vol. 36, No.1, pp. 27-36, 2004.

[12] K. D. Nguyen, I.-M. Chen, and T.-C. Ng, "Planning algorithms for S-curve trajectories," *Proc. of IEEE Int. Conf. on Advanced Intelligent Mechatronics*, pp. 1-6, 2007.

[13] P. Lambreshts, M. Boerlage, and M. Steinbuch, "Trajectory planning and feedforward design for electromechanical motion systems," *Control Engineering Practice*, Feb., vol. 13, pp. 145-157, 2005.

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

[15] 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, 1990.