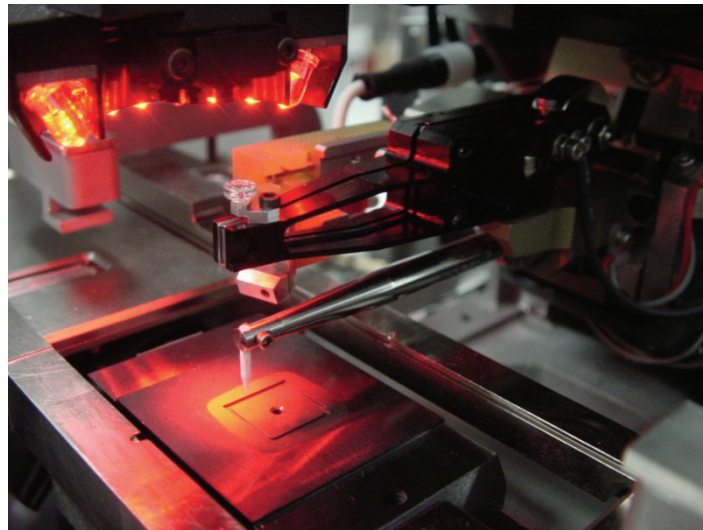


## Design of a Motion Profiles – Part B

- 3.1 Determination of the Motion time
- 3.2 Constraints on Motion Time
- 3.3 Calculation Example



## 3.1 Determination of the Motion time

모션 프로파일의 설계 후 스테이지stage의 실제 구동을 위해서는 주어진 거리를 어느 정도의 시간 안에 구동을 끝마칠 것인지 결정하여야 한다. 스테이지를 천천히 구동해야 할 경우는 큰 문제가 없지만, 만약 가능한 가장 빠른 시간 안에 스테이지를 구동하려고 한다면, 주어진 거리에서 그 시스템에서의 최초 구동시간의 한계를 구하고 이보다 더 짧은 시간에 구동되는 것을 막아 주어야 한다.

점대점point to point 구동의 경우 반도체 장비나, 생산에 관련된 장비들은 정해진 거리에 대한 구동 시간이 빠르면 빠를수록 좋은 경우가 많다. 설계자는 주어진 설계 파라미터 이내에서 가능한 한 가장 빨리 구동하게끔 하고 싶을 것이다. 특정 시스템에서 정해진 거리에 대한 최소 구동시간은 몇 가지 시스템의 한계에 의해 결정되는데, 특정 거리에 대한 제한조건이 전부 다 만족되도록 구동 최소 시간이 설계되어야 한다.

스테이지의 최소 운동시간을 결정짓는 가장 대표적인 파라미터 3가지는 시스템의 최대 허용 속도allowable maximum velocity, 최대 허용 가속도allowable maximum acceleration 및 최대 허용 저크allowable maximum jerk이다. 각각의 파라미터가 어떻게 구동 시간의 결정에 어떻게 영향을 미치는지 단계별로 알아보기로 하자.

최소구동시간의 계산에 앞서서, 먼저 서보 시스템의 구성에 대하여 간략하게 살펴보자. 다음의 Fig. 3.1은 서보 시스템의 구동부분의 구성을 간략하게 도시한 개략도이다. Fig. 3.1에서는 리니어 모터 스테이지를 도시하였지만 일반 로터리 모터를 이용한 서보 시스템 또한 유사한 구조를 가진다. 정밀 구동시스템은 주로 전류제어 모드로 동작하는데, 이는 크게 모션제어기, 드라이버driver, 채배기 및 모터 등의 모듈로 구성된다.

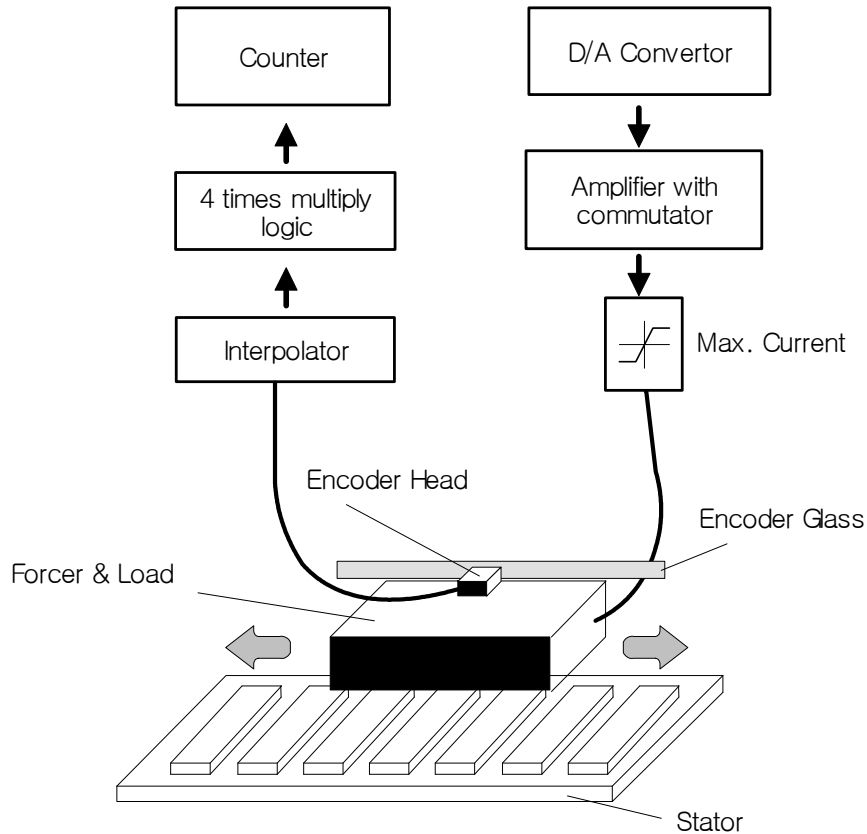


Fig. 3.1 Block diagram of a linear motor system

마이크로프로세서에서 계산된 제어기의 출력값은 D/A 컨버터를 거쳐서, 드라이버driver로 보내지게 된다. 보통 모터 드라이버는 전류제어 모드와 전압제어 모드가 있는데, 전압제어는 특수한 경우 사용되며, 보통의 위치 제어시스템에서는 주로 전류제어 모드가 이용된다. 따라서 전류제어 모드를 사용하는 경우 드라이버로 보내지는 명령은 전류에 대한 명령이므로 앞 절에서 설명한 가속도 프로파일과 유사한 프로파일의 명령이 입력된다.

드라이버driver는 전류에 대한 명령을 D/A 컨버터를 통해 전압 값으로 받아서, 독자적으로 내장하고 있는 전류 제어루프를 통해 전류 제어를 행하게 된다. 드라이버는 일반적으로 과전류 보호 장치가 내장되어 있으며 흘릴 수 있는 최대 전류가 정해져 있다. 또한 AC 서보나 브러시리스brushless 타입의 모터의 경우 커뮤테이션commutation이 필요 하므로 홀hall 센서나 엔코더를 이용하여 드라이버에서 이 작업을 수행해 준다.

엔코더encoder의 경우 채배기interpolator가 내장되어 있는 경우도 있으나, 고성능 리니어 엔코더의 경우, 정현파를 출력하는 헤드head부와 인터플레이터interpolator가 나누어져 있는 경우도 많다. 마지막으로 카운터의 입력 시 보통 4채배 로직을 통하여 해상도resolution를 4배 키워주는 것이 일반적이다.

지금까지 서보 시스템의 구동부에 대하여 간략하게 설명하였다. 구동부에 대한 좀 더 자세한 설명은 다음 섹션에서 각종 하드웨어 구성 테크닉과 함께 제공할 것이다. 그럼 다시 최소구동시간의 결정에 초점을 맞추어 위에서 설명한 최대 허용 속도, 가속도 및 저크를 결정짓는 요소들과, 이 세 가지에 전체적으로 영향을 미치는 위치의 다이내믹dynamic 에러요인에 대하여 분석해 보자.

### ❶ 최대 허용 속도

서보 시스템 구동의 최대 허용속도는 주로 엔코더의 최대 출력 주파수maximum output frequency와 엔코더 카운터부의 최대 입력 주파수maximum input frequency에 의하여 주로 결정된다. 하지만 엔코더의 채배기interpolator 가 별도로 부착되어 있는 경우, 정현파를 출력하는 엔코더 헤드head부, 정현파를 잘게 쪼개어 미세한 해상도를 얻어내는 엔코더 채배기interpolator 그리고 디지털 펄스로 출력되는 위치신호를 저장하는 카운터의 4채배 로직 구동 주파수 등 3가지 주요 모듈을 동시에 고려해야 한다. 다음의 3가지 사항은 서보 시스템에서 일반적으로 사용하는 리니어 및 로터리 엔코더의 설계시 반드시 고려해야 할 사항들이다.

#### (a) 엔코더의 헤드head부의 제한 속도

모션제어에서 많이 사용되는 광학식 엔코더는, 내부적으로 광 다이오드의 쌍으로 구성되

어 회전속도나 운동속도가 빨라짐에 따라 그 출력 신호의 크기가 줄어들기 때문에, 속도가 영zero일 때의 신호 크기를 기준으로 -3dB(선형 좌표계로 신호의 크기가 0.707배 되는 상태) 되는 주파수를 일반적으로 최대 출력 주파수로 규정하고 있다. 따라서 엔코더 스펙에서 확인할 수 있는 최대 회전수(혹은 최대속도)보다 더 빨리 구동하면, 광 슬릿을 통과한 수광부에서 발생하는 전류의 양이 작아져서 엔코더가 오동작하게 된다. 이는 광학식이 아닌 자기식 엔코더도 마찬가지이다.

엔코더 모듈을 출력에 의해서 분류하면, 일반적으로 TTL 방식등의 구형파가 출력되는 타입과, 정현파sinusoid가 출력되는 타입으로 나눌 수 있다. 구형파로 출력되는 방식의 엔코더는 내부적으로 필터와 비교기comparator를 장착하여, 펄스로 변환하여 출력해 주는 것이고, 정현파로 출력하는 타입은 뒷단에 해상도를 증폭하기 위한 채배기를 붙이도록 정현파를 출력하도록 되어 있다.

엔코더 헤드부의 제한속도는 매우 결정적인 변수이므로 구동 중 어떤 순간에도 이 속도를 초과하면 안 되며, 노이즈 등을 감안하여 10% 내외의 안전 영역을 두는 것이 좋다.

#### (b) 엔코더 채배기interpolator의 입출력 주파수의 한계

엔코더 채배기는 정현파의 sin, cos 신호를 잘게 나누어 해상도를 높여주는 역할을 하는데, 보통 20, 32, 50, 64, 128, 256 채배 등을 많이 이용한다. 이 채배기 또한 내부에 아날로그-디지털 변환기A/D converter, 고주파 클럭clock 그리고 광대역 OP 앰프 등으로 구성되어 있으므로 최대 허용 입력 주파수가 존재한다. 따라서 별도의 엔코더 채배기를 사용하는 경우, 스테이지 등이 최고 속도로 움직일 때 채배기의 최대 허용 입력 주파수를 넘지 않게 채배기를 선정하여야 한다. 마찬가지로 채배 계인을 바꾸어서 사용하는 경우, 채배기 자체의 최대 출력 주파수 또한 고려되어야 한다.

### (c) 카운터의 입력 주파수의 한계

기존의 A상, B상의 구형파 펄스의 출력을 가지는 모터의 경우, 일반적으로 DSP 보드나 제어기 보드에서 이를 정회전, 역회전을 구분하여 펄스를 누적하는 카운터(counter) 회로가 필요하다. 카운터는 EPLD(Erasable Programmed Logic Device)등으로 설계하는 것이 보편적이며, 보통 카운터 쪽의 입력부에 4채배 로직(logic)을 기본적으로 설계하여 이전 모듈에서 나온 펄스 신호를 마지막으로 4채배를 더 해주는 것이 현재의 서보 제어기에서의 일반적인 형태(configuration)이다. 따라서 EPLD의 동작 클럭 주파수가, 구동부가 최대 속도로 움직일 때의 채배기 출력 펄스 주파수의 4배 이상이 되어야 4채배 로직이 정상적으로 동작한다.

### ② 최대 허용 가속도

최대 허용 가속도는 주로 서보를 구동하는 모터가 발휘할 수 있는 최대 힘과 부하의 크기에 의해서 결정되므로, 결과적으로 부하, 모터상수 그리고 앰플리파이어(amplifier)가 낼 수 있는 최대 전류에 의하여 결정된다. 예를 들면 모터의 추력 상수가 1A당 10N이고, 모터 앰플리파이어의 최대 전류가 5A, 그리고 구동의 부하의 크기가 1kg이라면 마찰력을 무시한 최대 가속도는 단순히, 를 이용하여 다음과 같이 계산된다.

$$a_{\max} = 10(N/A) \times 5(A) / 1(kg) = 50m/s^2 \quad (3.1)$$

이때의 최대 가속도는  $50m/s^2$ 로 중력가속도로 환산하면( $1G=9.8m/s^2$ ) 약 5G 정도의 최대 가속을 가지게 된다. 이 경우 구동 시스템의 최대 능력이 5G인데 7G로 구동하라는 명령을 내린다면, 당연히 다이내믹(dynamic) 에러가 증가하며, 구동 성능이 저하될 수밖에 없다. 또한 부하의 계산시 기어나 각종 마찰에 대한 크기를 포함해야 하고, 에너지의 전달률을

포함한 손실계수 만큼의 설계 마진을 두어야 한다.

#### ③ 최대 허용 저크

최대 허용 저크의 결정은, 시스템의 제어 주파수(인터럽트 주파수), 모터 드라이버의 전류제어 밴드폭(current control bandwidth), 모터의 시정수(time constant), 모터의 공급 전압, 구동부의 기계적인 강성, 가이드의 강성, 구동부의 추력 중심과 무게 중심의 차이 등등, 제어 시스템 전체의 성능에 따라 결정되는 파라미터로서, 10G이상의 초고속 시스템의 경우 이 최대 허용 저크를 얼마나 높일 수 있느냐에 따라 그 시스템의 운동시간에 대한 성능이 좌우된다. 이 최대 허용 저크의 분석은 경험을 많이 필요로 하고, 진동특성을 고려하여 결정되고, 반드시 실험에 의하여 검증되어야 한다. 최대 허용 저크는 시스템의 전체적인 overall 성능지표라 볼 수도 있다.

#### ④ 최대 허용 다이내믹 에러(allowable dynamic error)

다이내믹 에러(dynamic error)는 서보 운동시 발생하는 시간에 따른 위치 오차를 말한다. 최대 허용되는 다이내믹 에러는 전체 시스템의 허용 오차를 결정 짓기 때문에 모듈별 스펙(specification)에도 큰 영향을 미친다. 위에서 정의한 각각의 최대 허용 파라미터를 넘어서는 값으로 시스템을 구동하면, 다이내믹 에러가 증가하는데, 이러한 최대 허용 에러의 크기는 앞에서 제시한 3가지 파라미터의 결정에 영향을 미친다.

지금까지 운동시간을 결정짓는 최대 속도, 가속도, 저크, 에러 등 4가지 요소에 대해서 알아보았다.

이 4가지의 요소 중 4번의 최대 허용 다이내믹 에러를 제외한 나머지 세 가지를 보통 실제 구동시간의 계산시 고려하는데, 3번의 저크 항목은 3차식의 구동 프로파일을 사용

하는 경우 고려할 수 없고, 5차 이상의 방정식을 사용할 경우 반영할 수 있다(이에 대해서는 다음 섹션을 참고). 그러면 어떠한 특정 거리가 주어졌을 때 실제로 이러한 요소를 어떻게 고려하여 구동시간을 정하는지 Design Example 을 통해서 생각해 보자.

### **Design example 3.1**

**3.1 장에서 제시한 3차의 구동 방정식의 위치 프로파일을 1000개의 테이블에 저장 하여 놓고 사용하며, 또한 4kHz의 인터럽트를 가지는 DSP를 이용한다고 가정하자. 제어하고자 하는 대상 플랜트는 플립칩 관련 반도체 검사 이송 테이블인데, 전자 및 기구 시스템을 모두 포함한 제어 대상계의 최대 허용 속도는 1000mm/s, 최대 허용 가속도는 30m/s<sup>2</sup>라고 분석되었다. 3.1 절의 3차식 모션 프로파일을 이용하여 설정거리에 따른 최소 구동시간을 계산하는 알고리즘을 구현하고 100mm를 구동할 경우의 최소 구동시간을 계산하라.**

<Solution 3.1>

테이블을 이동한 구동에서, 최종적으로 매 인터럽트interrupt당 가야 할 거리는, 주어진 테이블을 매 인터럽트마다 일정 간격으로 읽어서 운동 거리에 곱하여 계산해 낼 수 있다. 예를 들면 5mm를 10ms에 이동한다면, 인터럽트가 4kHz 이므로 10 ms 동안 40번의 제어 출력을 이용하여 목적지인 5mm 전방으로 이동하는 것이 되겠다. 따라서 1000개의 위치 테이블을 40번으로 나누어 25개의 간격을 두고 매 인터럽트마다 표준화된normalized 위치 테이블을 읽어서 가고자 하는 거리를 곱해 주면 그때의 목표 위치를 구해낼 수 있다. 이러한 방법을 도식화하면;

필요한 인터럽트의 수 = 구동시간 × 인터럽트 주파수

테이블 참조 인덱스 폭width= 테이블의 셀cell 수 / 구동시 필요한 인터럽트의 수



등으로 정리 할 수 있겠다.

만약 주어진 거리 100mm를 구동한다고 했을 때, 테이블 포인터의 값을 작게 결정한다면 여러 번의 인터럽트가 필요하므로 구동시간이 길어지게 된다. 반대로 매 인터럽트 시 테이블을 널찍널찍하게 건너 뛰어 읽는다면 그만큼 빨리 최종 목적지에 도착하게 되므로 구동시간이 짧아지게 된다.

다시 문제의 본질로 돌아와서 그렇다면 어떻게 구동 시간을 결정할 것인가를 살펴보자. 허락하는 한 가장 빠른 시간을 구하고자 한다면 현 모션 구동 프로파일이 제어 대상계의 최대 허용 속도, 가속도중 어떤 한계에 걸리는지 알아내야 한다.

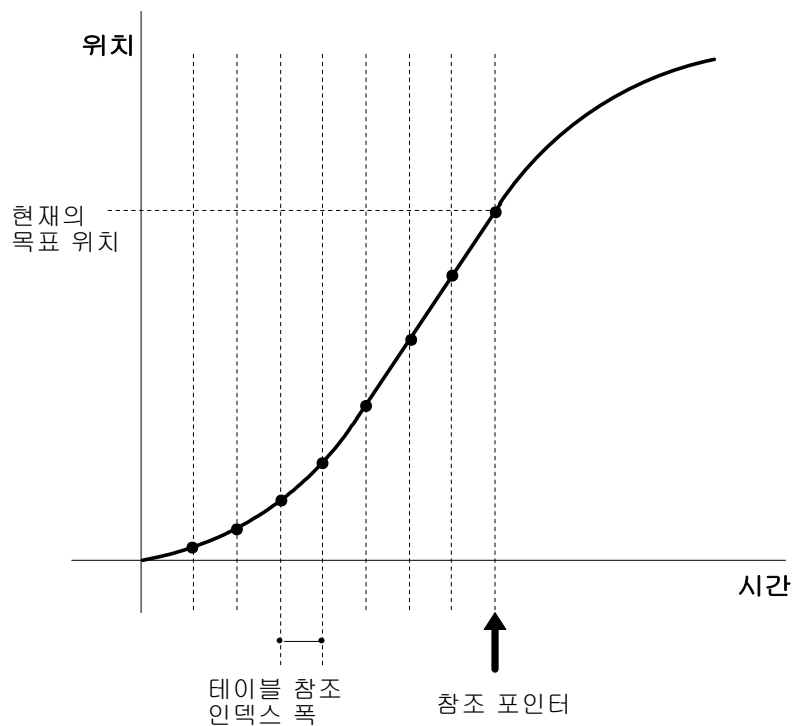


Fig. 3.2 Movement of reference in Motion Table

먼저 최대 속도를 고려하여 보자. 앞에서 구한 3차 방정식에 의한 구동 프로파일 중, 속도 프로파일을 보면  $t=T/2$ 일 때 최대 속도가 발생하는데, 이때의 최대 속도가 시스템이 허용하는  $1000\text{mm/s}$ 를 초과하지 못하는 구속조건이 발생한다. 식(2.13)에서 최대 속도는  $t=T/2$ 에서 발생하므로 이때의 최대 속도를 구하면,

$$\frac{1}{S} \times \dot{y}_{\max}(0.5) = \frac{6}{T_v}(0.5) - \frac{6}{T_v}(0.5)^2 = \frac{1.5}{T_v} \quad (3.2)$$

식 (3.2)를 다시 정리하면, 최대속도에 의한 제한속도  $T_v$ 는,

$$T_v = 1.5 \times \frac{S}{\dot{y}_{\max}} \quad (3.3)$$

가 된다. 여기서 시스템의 최대 속도 능력이  $1000\text{mm/s}$  라 하였으므로,  $100\text{mm}$ 를 이동할 때의 최대 허용속도  $1000\text{mm/s}$ 에 대한 최소 구동시간은,

$$T_v = 1.5 \times \frac{100\text{mm}}{1000\text{mm/s}} = 150\text{ms} \quad (3.4)$$

으로 구하여진다. 결과에 대하여 부연설명을 하자면, 만약  $100\text{mm}$  거리를  $150\text{ms}$  보다 작은 시간으로 구동한다면,  $T/2$  시점에서 최대 허용 속도 능력인  $1000\text{mm/s}$ 를 초과하게 되고, 이는 엔코더 및 카운터 시스템에 무리를 주어 엔코더 오동작에 의한 시스템 폭주가 일어날 가능성이 매우 커진다. 따라서 위의 시스템에서  $100\text{mm}$  거리를 움직일 때의 최대 허용 속도에 의한 최소 구동 시간은  $150\text{ms}$  이므로  $100\text{mm}$  거리를 구동할 때는 적어도  $150\text{ms}$  이상의 시간을 주어야 한다.

위에서 구한 것은 속도에 대한 한계이며, 이번에는 시스템이 갖는 최대 허용 가속도에

의한 한계를 구하여 보자. 속도에 의하여 구할 때와 마찬가지로 가속도에서의 한계값 또한 유사하게 구하여 진다. 가속도에 대한 식 (2.14)에서 최대 가속도는 양 끝단에서 발생하므로  $t=0$ 을 대입하면,

$$\frac{1}{S} \times \ddot{y}(0) = \frac{6}{T_A^2} \quad (3.5)$$

식 (3.5)를 다시 정리하면,

$$T_A = \sqrt{\frac{6S}{\ddot{y}_{\max}}} = \sqrt{\frac{6 \times 0.1m}{30m/s^2}} = 141.42ms \quad (3.6)$$

로 구하여 진다. 여기서 동일거리 100mm에 대하여, 최대속도제한에 의한 최소구동시간은  $= 150ms$ , 최대가속도제한에 의한 최소구동시간은  $= 141.4ms$  으로 각기 다르게 구하여 진 것을 알 수 있다. 따라서 실제 구동시 이 두 가지 조건을 다 만족하여야 하므로 둘 중 느린 시간이 최소구동시간minimum running time이 된다. 따라서 위 시스템에 대한 100mm거리에 대한 최소 구동시간은 둘 중 느린 구동 시간인 150ms 가 된다. 따라서 이 구동길이(100mm)에서의 구동 시간의 병목은 바로 시스템의 최대허용속도에 있으며, 역으로 이 구동길이에 대한 시간을 단축시키기 위해서는 시스템의 최대허용속도를 향상시켜야 한다는 사실 또한 알 수 있다. 따라서 설계하는 장비나 시스템이 이 거리에서의 구동이 주류를 이루는 작업을 한다면, 엔코더 및 채배기의 성능을 개선하여 최대 구동 속도를 향상시켜야 한다.

<End of solution 3.1> ■

위의 3차식 프로파일에 대한 최소구동시간의 결정을 도식화하여 도표로 나타내면 다음의 Fig. 3.2와 같다.

구동 프로파일과 최소구동시간 등은 바로 그 시스템의 구동 스펙specification이 되어 전체 시스템의 시간당 생산량이나 기타 이송 능력 등에 바로 영향을 주므로, 서보 시스템에 사용되는 모듈의 선정 및 설계는 프로젝트의 초기 개발 단계에서 전체 시스템의 목표에 의해 결정되고, 검증되어야 한다. 그리고 위의 Design Example 3.1의 경우에는 제어 대상 시스템의 최대허용속도와 최대허용가속도만 고려했으므로 두 가지 조건에 대해서 다 만족하는 시간을 구하면 되지만, 만약 최대허용 저크jerk 항 까지 고려하는 경우에는 3가지 조건을 다 만족하는 시간을 구하여야 한다. 또한 위에 구한 식 (3.3)과 (3.6)등은 3차 방정식의 구동 프로파일에 대한 것이므로 만약 다른 종류의 프로파일을 사용한다면, 마찬가지로 방법으로 대상 프로파일에 맞는 계산식을 구해주면 된다.

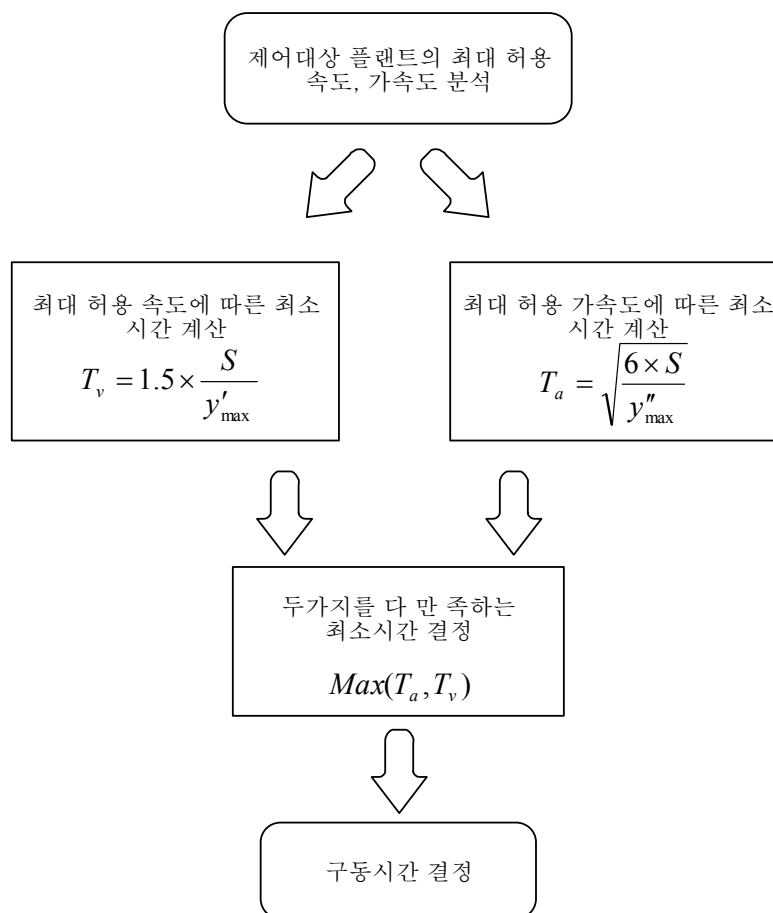


Fig. 3.3 Determination of the Motion Time

구동시간의 결정은 이와 같은 시스템의 제약조건을 확인하고, 구동 프로파일에 의하여 결정하게 되므로, 서보 시스템을 이용한 생산 장비의 단위시간당 생산량 등을 분석하기 위해서는 구동 시간의 결정에 영향을 미치는 파라미터를 선행해서 분석하고 결정하여야 한다.

3차식에 의한 구동 프로파일의 거리별 최소구동시간을 구해보면, 짧은 거리에서는 최대 허용가속도가 병목bottle neck이 되고, 조금 긴 거리에서는 최대허용속도가 병목이 된다. 다음의 Matlab Source 3.1은 3차 구동 방정식의 구동 시간을 계산하여 도시하여 주는 프로그램이다. Fig. 3.4는 거리에 따른 최소구동시간을 구하여 도시한 것이다(시스템의 최대 허용속도 1000mm/s, 최대허용가속도  $30\text{m/s}^2$  일 경우).

### Matlab Source 3.1

**% 3th order motion time calculator**

**% Number of Table cell is 2048**

**% HanKim**

**clear;**

**MA=input('\n\n Enter the maximum acceleration(m/s^2): ');**

**MV=input('\n\n Enter the maximum velocity(mm/s): ');**

**index=1;**

**flag=0;**

**cstep=100;**

**final\_dist=300000;**

**fx=(1:cstep:final\_dist)';**

**for x=1:cstep:final\_dist,**

**Tv=(1.5\*x)/MV;**

**Ta=2.4495\*sqrt(x/MA);**

**if Tv >= Ta**

```

    flag = 1;
    TF=Tv;
else
    flag=2;
    TF=Ta;
end
Svel(index) = 1.5*x/TF;
Sacc(index) = 2.4495^2*x/TF^2;
SVfg(index)=flag;
SVtime(index)=TF;
index = index + 1;
end

figure(1);plot(fx,SVtime,'b-');grid on;
title('Working Time');xlabel('distance um');ylabel('time ms');
figure(2);plot(fx,Svel,'b-');grid on;
title('Max Velocity variation with distance');xlabel('distance um');ylabel('mm/s');
figure(3);plot(fx,Sacc,'b-');grid on;
title('Max Acc variation with distance');xlabel('distance um');ylabel('m/s^2');

```

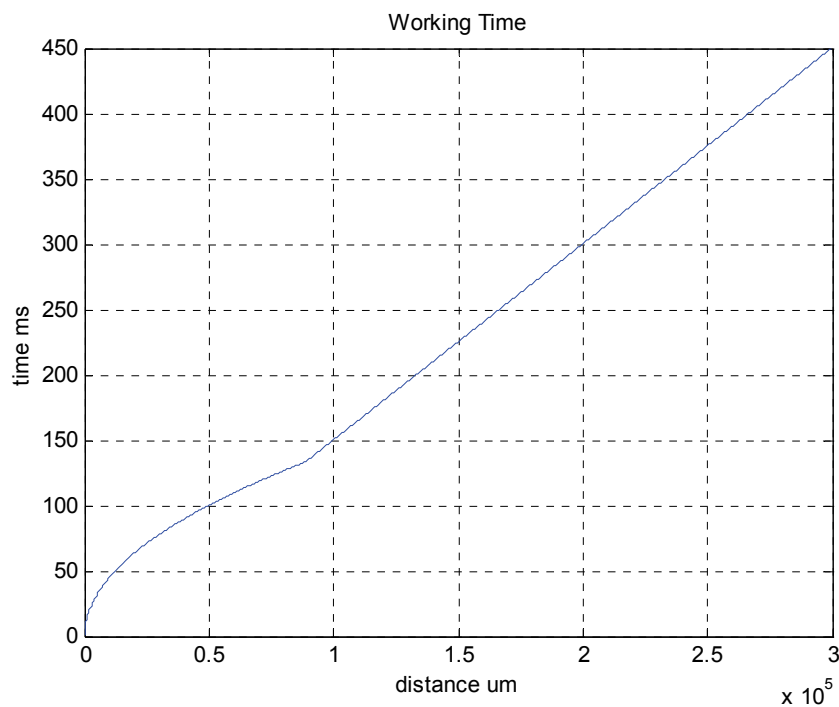


Fig. 3.4 Motion Time according to motion distance,

$$(V_{\max} = 1000\text{mm/s}, A_{\max} = 30\text{m/s}^2)$$

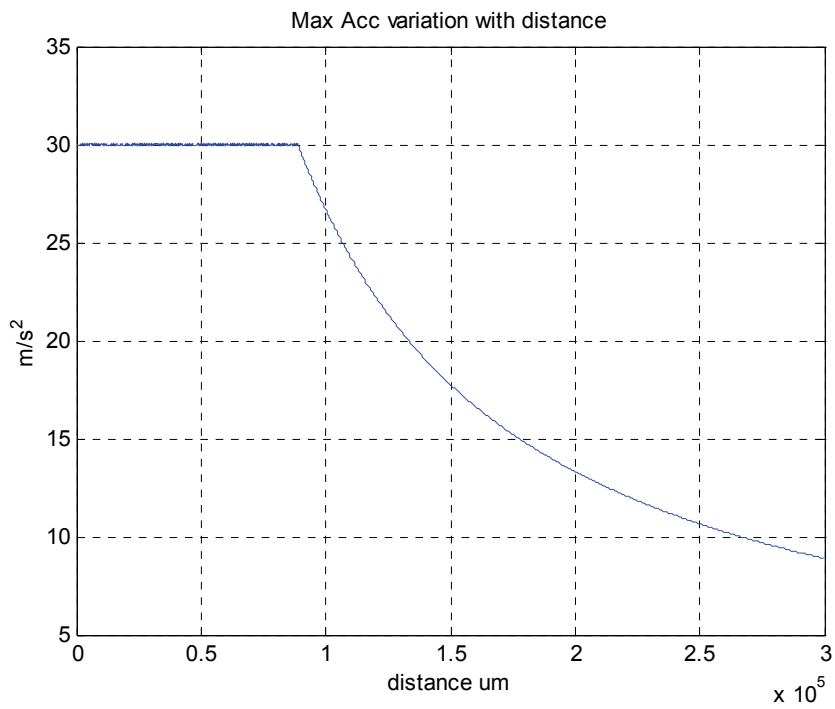


Fig. 3.5 Max Acceleration according to motion distance,  
( $V_{\max} = 1000\text{mm/s}, A_{\max} = 30\text{m/s}^2$ )

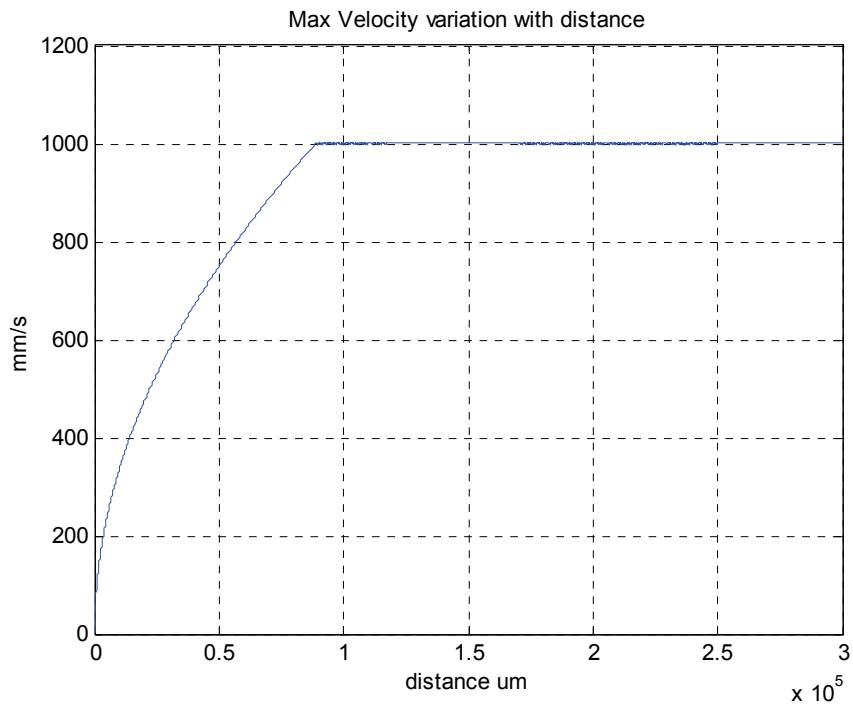


Fig. 3.6 Max Velocity according to motion distance,

$$(V_{\max} = 1000\text{mm/s}, A_{\max} = 30\text{m/s}^2)$$

위의 Fig. 3.4에서 보면, 구동 거리로 대략 85mm 부근에서 최소 시간이 정해지는 병목 구간이 바뀌는 것을 알 수 있다. 대략 85mm를 기점으로 하여, 이것보다 짧은 거리는 최대허용가속도에 의한 제한이 걸리다가 85mm 보다 먼 거리를 구동할 때는 최대허용속도에 의한 제한이 걸리는 것을 알 수 있다.

상기의 경우, 자신이 설계하는 서보시스템의 주요 구동 범위가 주로 85mm 보다 작은 스트로크를 가지는 경우에는 위에서 설계한 프로파일을 그냥 사용해도 좋지만, 만약 주된 이동 거리가 85mm를 훨씬 넘는다면 위의 3차식의 프로파일은 사다리꼴의 프로파일에 비해 느린 구동시간을 제공할 것이다. 다음의 그림을 보자.

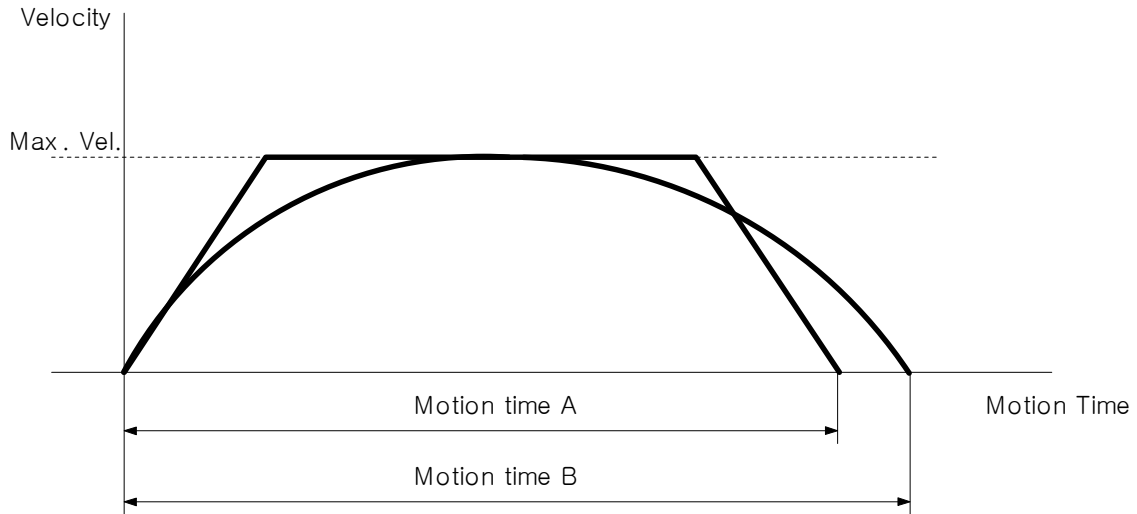


Fig. 3.7 Trapezoidal vs. 3<sup>rd</sup> order Profiles

같은 최대 속도 한계를 가질 때, 사다리꼴 속도 프로파일 같은 경우 최대 속도를 한동안 유지하지만, 3차식의 프로파일에서 유도된 속도 프로파일은 최대 속도지점이 꼭지점으로 나타나므로, 이를 적분했을 때의 면적이 같아지려면 3차식의 프로파일이 더 많은 시간을



소비한다. 따라서 구동거리가 길수록 3차 방정식에 의한 프로파일이 구동시간 면에서 뒤쳐지게 된다. 따라서 이러한 경우 최대허용가속도 쪽에 더 여유가 있다면, 3차식의 반을 잘라 등속구간을 삽입하는 알고리즘을 추가로 사용하면, 양쪽의 장점을 골고루 이용하는 셈이 된다. 다음의 그림을 보자.

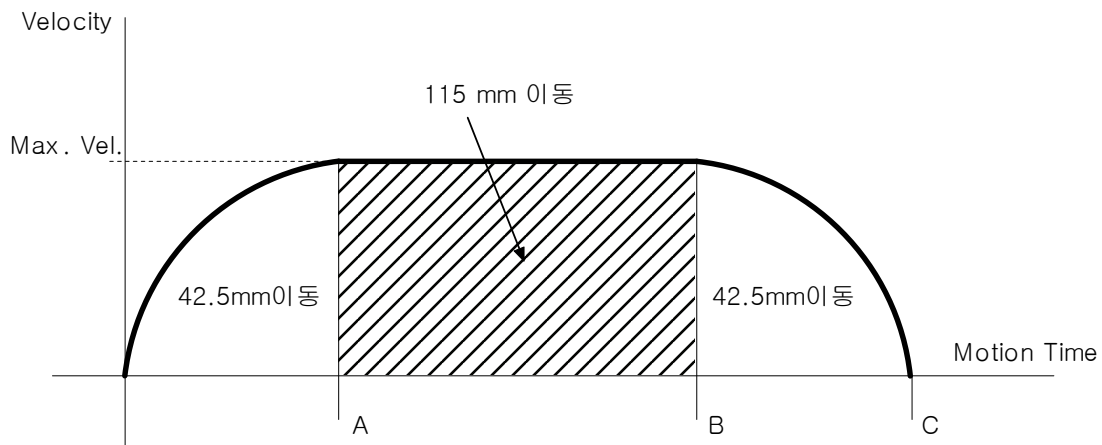


Fig. 3.8 3<sup>rd</sup> order Profile with constant velocity area

위와 같은 방법을 사용하면, 최대 속도 지점의 변곡점에서의 부드러운 속도 변환으로 기구적인 진동도 억제하고, 구동 속도 또한 사다리꼴에 비하여 비슷한 구동 프로파일을 설계할 수 있다. 이렇게 중간에 등속시간을 삽입한 경우 구동 시간을 결정하는 방법을 생각해 보자.

먼저 등속구간이 없는 기본 3차 프로파일에서 주어진 시스템에서 속도 한계로 넘어가는 지점을 파악한다. 위의 경우 85mm가 되는데, 이 한계점은 미리 계산상으로 알 수 있으므로 이를 이용하면 다음과 같이 손쉽게 구동시간을 계산할 수 있다.

- ❶ 만약 85mm 보다 작은 거리의 구동 명령이 들어오면, 등속구간을 삽입하지 않고 기본 3차 프로파일로 구동한다.

② 만약 85mm 보다 큰 거리의 구동 명령이 들어오면, 85mm를 초과하는 거리 동안만 등속으로 구동하면 된다. 예를 들어 200mm를 구동할 경우, 최대속도를 이용하여 115mm를 구동하면 되므로 3차식 프로파일의 반을 잘라서 최대 속도에 도달했을 때 이 속도로 구동하는 시간을 계산하여 등속구간을 삽입한다.

구동거리가 길어질 경우 등속구간이 없으면, 최소구동시간이 너무 길어지는 경향이 있으므로 많은 경우 상기와 같이 등속구간을 삽입해 주는 알고리즘을 삽입하여야 한다. 이때 모드 변화가 일어나는 변곡점에서의 이산화 에러(quantization error)를 최소화 하도록 코딩(coding)해야 한다.

예를 들면, 상기의 경우 115mm의 거리를 최대허용속도 1000mm/s로 구동한다고 할 때, 엔코더 1펄스가 0.3mm, 인터럽트 주기가 3ms 라고 가정을 해 보자. 먼저 인터럽트 변수는 정수로 이루어지므로 이에 대한 영향을 고려해 보자. 1000mm/s의 속도로 115mm를 구동한다면, 등속을 유지해야 하는 시간은,

$$T_{cv} = 115/1000 = 115ms \quad (3.7)$$

로 계산된다. 그런데, 인터럽트 주기가 3ms 이므로 이를 인터럽트 발생 시간으로 나누면,

$$N_i = 115/3 = 38.33 \quad (3.8)$$

로 정수로 나누어지지 않는 경우가 대부분이다. 또한 엔코더의 펄스도 정수이므로 수식으로 계산된 속도를 구현하려할 때 같은 문제가 생긴다. 따라서 많은 경우, 등속모드가 바뀔 때의 변곡점에서 비연속적인 속도 목표치가 생성되거나 잘못하면 목표 위치에 약간 덜 가거나 더 지나가는 현상마저 일어날 수 있다. 이 두 가지 문제는 여러 가지 아이디어로 해결할 수 있는데, 다양한 방법이 있으므로 독자 여러분의 몫으로 남긴다.

정리하면 이렇게 구동 프로파일은 해당 거리에 대한 최소구동시간과 밀접한 관계를 가지고 있기 때문에, 서보 시스템 엔지니어는 설계시 구동시간에 제약을 가하는 요소들을 숙지하여, 그 시스템에 최적인 구동 프로파일을 설계 적용하여야 하며, 등속구간의 삽입은 다음에 소개되는 구동 프로파일에도 똑같이 적용되므로, 앞으로 소개되는 구동프로파일에도 필요할 경우, 설계자가 제어대상 플랜트와 주요 구동거리에 따라 위에서와 동일하게 중간에 등속구간을 삽입하여 최소구동시간을 줄여주어야 한다.

