

Stepping Motor Control 결과보고서

20101449 한가은

목차

| | |
|---|----|
| 1. 서론 | 3 |
| 1.1. 원리 및 이론 | 3 |
| 1.2. 실험 배경 및 조건 | 4 |
| 2. 실험 결과 및 분석 | 5 |
| 2.1. GPIO를 이용한 스텝핑 모터 구동 | 5 |
| 2.2. 스텝핑 모터 구동 API 작성 | 6 |
| 2.3. 속도 프로파일의 적용 | 8 |
| 2.3.1. 사다리꼴 프로파일 | 8 |
| 2.3.2. 삼각형 프로파일 | 11 |
| 2.4. Stepping Motor 고속 제어를 위한 가속도 및 속도 튜닝 | 14 |
| 2.4.1. 가속도 튜닝 | 14 |
| 2.4.2. 속도 튜닝 | 15 |
| 3. 결론 | 16 |
| 3.1. 오차 분석 | 16 |
| 3.2. 결과 고찰 | 16 |

1. 서론

1.1. 원리 및 이론

Stepping motor란 일정한 각도 단위로 회전하는 모터이다. 전기 회로를 이용하여 권선 전류의 방향을 바꿔 회전을 제어하며, 통상적으로 오픈루프 제어기법을 사용한다.

Stepping motor 제어 시 회전자가 권선 전류 변화 속도를 따라가지 못해 제자리에 진동하는 탈조 현상을 유의해야 한다. 특히, 통상적으로 오픈루프 제어기법이 사용되기에 탈조 현상을 감지할 수 있는 센서가 별도로 사용되지 않기에 제어 기법을 통해 탈조 현상을 방지해야 한다.

탈조 현상을 방지하는 제어 기법으로는, 가감속 제어가 해당된다. 이는 급격한 가속도나 속도가 발생하지 않도록, 매 스텝 이동함에 따라 상전환 시간을 조금씩 감소시키는 제어 기법이다.

Full step 구동 방식에 대하여, Stepping motor의 위치 제어 및 속도 제어는 스텝 각 $\Delta\theta$, 스텝 수 k , 상 지연시간 Δt 에 대해 다음의 수식으로 나타낼 수 있다.

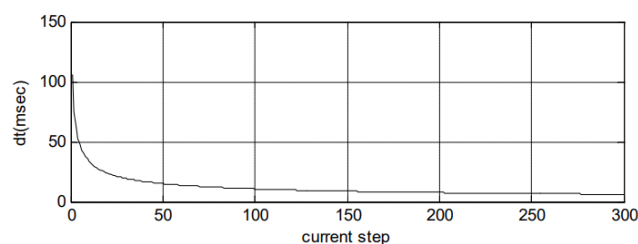
$$s = \Delta\theta \times k$$

$$v = \Delta\theta / \Delta t$$

또한, 급격한 가속도나 속도가 발생하지 않는 등가속도 회전을 만드는 각 스텝 사이 상 지연 시간 Δt 는 다음과 같이 나타낼 수 있다.

$$\Delta t = \sqrt{\frac{\Delta\theta}{2ak}}$$

위 수식으로 나타나는 Δt 를 스텝 수 k 에 대해 그래프로 나타내면 다음과 같다.



1.2. 실험 배경 및 조건



[그림 1: 실험에 사용된 기구]

본 실험에는 [그림 1]의 기구를 사용하였다. Stepping motor에는 바퀴 부하가 달려있으며, 실험에 사용된 Stepping motor는 4상 하이브리드형으로, 1.8 degree의 분해능을 갖는다.

정방향을 시계방향(CW), 역방향을 반시계방향(CCW)로 설정하였으며, 가감속 제어 기법을 위해 사다리꼴 속도 프로파일을 적용하였다. 이 때, 사용한 timer interrupt frequency는 100,000 Hz이다.

또한, 사다리꼴 속도 프로파일 적용 시 짧은 거리 이동의 예외 상황을 고려하여, 만약 이동할 거리의 절반이 가속 구간에서 이동할 거리보다 짧다면 등속 구간이 없는 삼각형 속도 프로파일을 사용하였다.

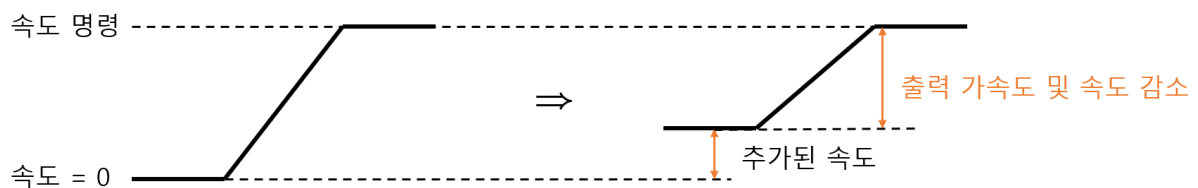
2. 실험 결과 및 분석

2.1. GPIO를 이용한 스텝핑 모터 구동

Full step 구동에서 모터가 더 이상 회전하지 않을 때까지 delay time을 감소시켜 본 결과, 0.00145보다 작은 delay time에서 탈조 현상이 발생하는 것을 확인하였다.

Full step minimum delay time: 0.00145 (초)

이때 손으로 모터를 회전시키고 나면 정상적으로 회전이 잘 이루어지는 것을 확인할 수 있었고, 그 이유는 속도를 부여함으로써 따라가야 하는 속도 크기를 줄였기 때문임을 알 수 있었다. 즉, stepping motor에서는 토크의 한계로 인한 가속도의 한계 및 속도의 한계로 탈조가 발생하는데, 손으로 속도를 주었기에 모터가 출력해야 하는 가속도 및 속도가 줄어들어 회전할 수 있게 된 것이다. 이 변화를 그래프로 나타내면 다음 [그림 2]와 같다.



[그림 2: 손으로 속도를 부여한 뒤 발생한 속도 및 가속도 변화]

주어진 skeleton code를 half step 구동 방식으로 바꾸어 최소 delay time을 확인한 결과, 0.00067초로 확인되었다. 이에 Full step 구동 방식과 Half step 구동 방식에서 속도 공식을 통해 계산한 각 최대 속도는 다음과 같다.

$$\text{Full step에서의 최대 속도} = \frac{\Delta\theta}{\Delta t} = \frac{1.8}{0.00145} = 1241 \text{ [deg/s]}$$

$$\text{Half step에서의 최대 속도} = \frac{\Delta\theta}{2\Delta t} = \frac{1.8}{2 * 0.00067} = 1343 \text{ [deg/s]}$$

오차가 발생하여 두 구동 방식에서의 최대 속도가 완벽히 일치하진 않지만, 비슷한 속도 부근에서 탈조 현상이 발생함을 알 수 있었다.

또한, Full step과 Half step 구동 방식에서 모터 1회전(360도)에 몇 스텝이 필요한지 이론적인 계산은 다음과 같다.

$$\text{Full step에서 스텝 수 } k = \frac{360}{1.8} = 200$$

$$\text{Half step에서 스텝 수 } k = \frac{360}{1.8/2} = 400$$

실험적으로 각 구동 방식에서 360도를 회전시키고, 측정한 필요 스텝 수는 다음과 같다.

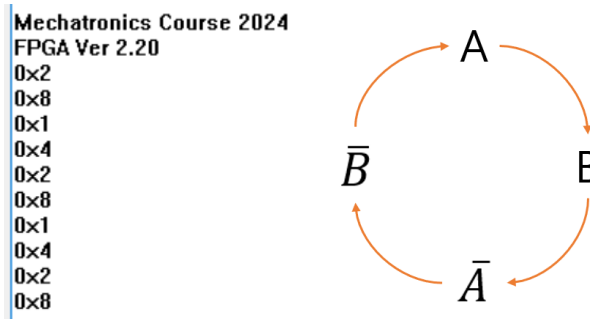
Mechatronics Course 2024
FPGA Ver 2.20
Full step: 200 steps per 360 deg
Half step: 400 steps per 360 deg

[그림 3: 각 구동 방식 별 모터 1회전에 필요한 스텝 수]

실험 결과, 수식으로 계산한 모터 1회전 필요 스텝 수와 일치한다는 것을 확인할 수 있었다.

2.2. 스텝핑 모터 구동 API 작성

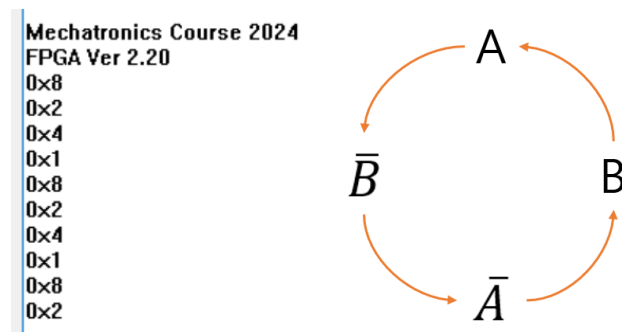
스텝핑 모터를 정방향 또는 역방향으로 1 스텝 회전하는 함수 *OneStepMove()*를 작성하여 정방향에 대해 함수를 10회 호출한 결과는 다음과 같다.



[그림 4: 정방향에 대해 *OneStepMove()* 10회 호출]

모터의 4상 A, B, \bar{A} , \bar{B} 에 따라 레지스터 값은 순서대로 0x2, 0x8, 0x1, 0x4이다. [그림 4]의 결과에서 10회 회전 시 $A \rightarrow B \rightarrow \bar{A} \rightarrow \bar{B} \rightarrow A \rightarrow B \rightarrow \dots$ 의 순서가 반복되는 것을 통해 정방향 회전이 정상적으로 이루어지는 것을 알 수 있다.

역방향에 대해 함수를 10회 호출한 결과는 다음과 같다.



[그림 5: 역방향에 대해 *OneStepMove()* 10회 호출]

[그림 5]의 결과에서 10회 회전 시 $B \rightarrow A \rightarrow \bar{B} \rightarrow \bar{A} \rightarrow B \rightarrow A \rightarrow \dots$ 의 순서가 반복되는 것을 통해 역방향 회전이 정상적으로 이루어지는 것을 알 수 있다.

다음으로, 회전시킬 각도와 속도를 argument로 전달받아 모터를 구동하는 *StepMoveCV()* 함수를 작성하고, 그 동작을 평가하였다. 속도를 바꿔보며 함수를 실행시켜보던 중, 180 deg/s 부근의 속도 명령이 발생하면 정상적으로 동작하지 않는 모습을 보였다. 앞선 실험 단계로부터 1상 여자 방식 full step 구동에서의 최대 속도가 1241 deg/s임을 판단했기에 탈조 현상이 아니라는 것을 의심할 수 있었고, 이에 속도를 더 낮춰 본 결과 90 deg/s 이하의 속도에선 다시 정상적으로 회전하는 모습을 보였다. 따라서 해당 실험 조건에서 모터는 90 deg/s ~ 180 deg/s 사이의 속도에서 공진 주파수의 영향으로 회전자가 정상적으로 회전하지 않음을 판단하였다.

2.3. 속도 프로파일의 적용

스테핑 모터의 고속 회전을 위하여 사다리꼴 속도 프로파일을 적용하고자 두 함수를 작성하였다.

- MakeVelProfile(): 각 스텝간 시간 지연을 타이머 인터럽트 발생 횟수로 변환하여 록업테이블을 생성한다.
- StepMoveVP(): 회전 각도, 최대 속도, 가속도를 입력받아 모터를 구동시킨다.

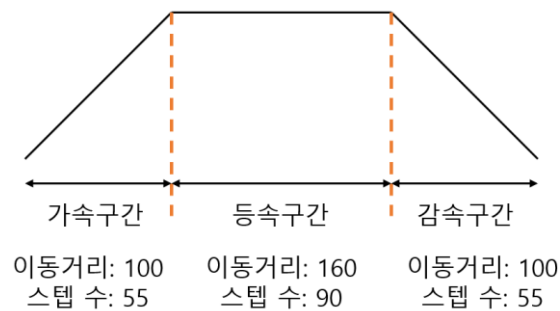
2.3.1. 사다리꼴 프로파일

MakeVelProfile() 함수가 정확한 록업테이블을 생성하는 지 확인하였다. 회전 각도 360도, 최대 속도 2000, 가속도 20000를 명령했을 때, 발생해야 하는 사다리꼴 속도 프로파일을 수식으로 계산한 결과는 다음과 같다.

$$\text{가속 구간 회전 각도 } S_1 = \frac{v_{max}^2}{2a} = \frac{2000^2}{2 \cdot 20000} = 100, \text{ 가속 구간 스텝 수 } k_1 = \frac{S_1}{\Delta\theta} = \frac{100}{1.8} \approx 55$$

$$\text{등속 구간 회전 각도 } S_2 - S_1 = S_{ref} - 2S_1 = 360 - 20 = 160, \text{ 등속 구간 스텝 수 } k_2 = \frac{S_2 - S_1}{\Delta\theta} = \frac{160}{1.8} \approx 90$$

$$\text{감속 구간 회전 각도 } S_{ref} - S_2 = 360 - 260 = 100, \text{ 감속 구간 스텝 수 } k_3 = \frac{S_{ref} - S_2}{\Delta\theta} = \frac{100}{1.8} \approx 55$$



[그림 6: 360deg, 2000 deg/s, 20000 deg/s²에서의 사다리꼴 속도 프로파일]

MACRO_PRINT 함수를 이용하여 각 구간 별 인터럽트 발생 횟수를 확인한 결과, 다음의 결과를 얻었다.

가속구간

등속구간

감속구간

| | | | | | | | | | | | | | | |
|---------|-----|-----|-----|-----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|
| step | 1 | 2 | 3 | ... | 56 | 57 | ... | 145 | 146 | 147 | ... | 198 | 199 | 200 |
| 인터럽트 개수 | 670 | 474 | 387 | ... | 90 | 90 | ... | 90 | 90 | 91 | | 387 | 474 | 670 |

- 55개 스텝의 가속 구간: 첫 구간의 인터럽트 발생 횟수는 670으로, 55번째 스텝에는 90까지 인터럽트 발생 횟수가 줄어드는 양상을 보였다.
- 90개 스텝의 등속 구간: 56번째 스텝부터 145번째 스텝까지는 일정한 인터럽트 발생 횟수 90을 유지하였다.
- 55개 스텝의 감속 구간: 146번째 스텝의 90개 인터럽트 횟수에서 200번째 스텝에는 670개로, 인터럽트 발생 횟수가 늘어나는 양상을 보였다.

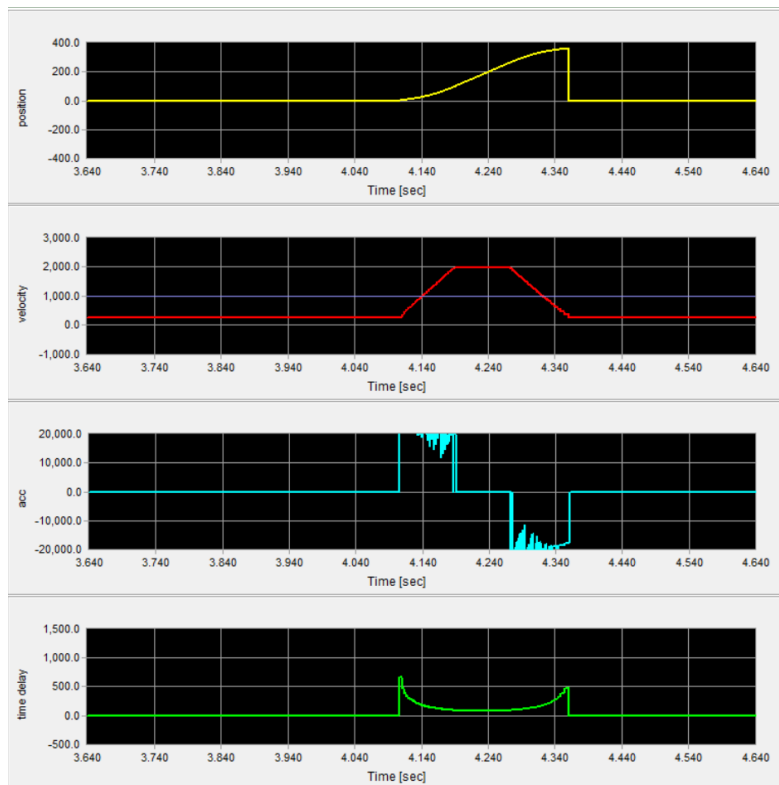
실험 결과, 인터럽트 발생 최소 개수는 90으로, 100,000 Hz 주기의 타이머 인터럽트를 사용할 때 속도는 다음과 같다.

$$v = \frac{\Delta\theta}{90/100000} = 2000$$

즉, 인가한 속도 명령과 같은 최대 속도가 발생하는 것을 알 수 있으며, 수식으로 계산한 사다리꼴 속도 프로파일과 동일한 모양이 만들어지는 것을 확인할 수 있다.

USB 모니터로 파형을 출력한 결과는 다음과 같다.

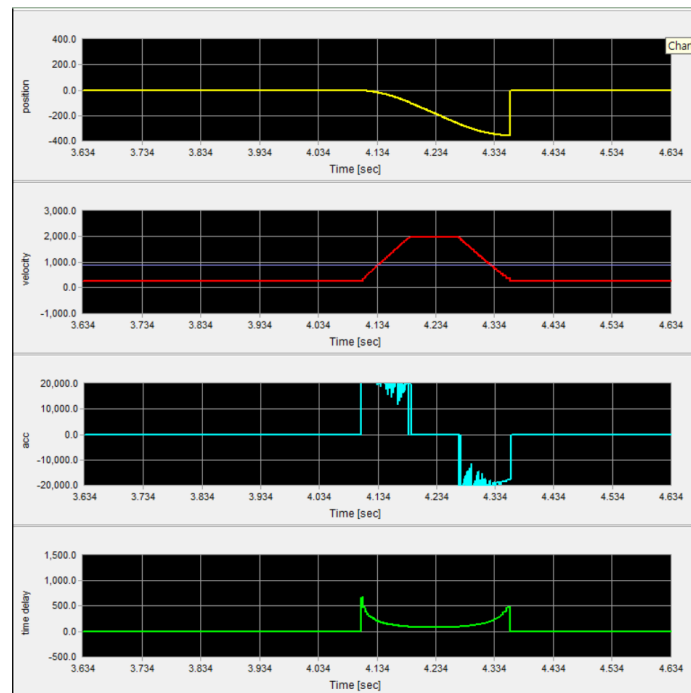
Stepping Motor Control



[그림 7: 360deg, 2000 deg/s, 20000 deg/s²에서의 USB Monitor 출력 파형]

출력 파형은 차례대로 모터의 위치, 속도, 가속도, 각 스텝 간 상 지연 시간이며, 정상적으로 가감속 제어가 이루어졌음을 확인할 수 있다.

또한, StepMoveVP() 함수의 동작을 검증하고자 같은 조건에서 CCW 방향으로, 즉 -360도를 명령한 결과 USB Monitor 출력 파형은 다음과 같다.



[그림 8: -360deg, 2000 deg/s, 20000 deg/s²에서의 USB Monitor 출력 파형]

마찬가지로 출력 파형은 차례대로 모터의 위치, 속도, 가속도, 각 스텝 간 상 지연 시간이며, 정상적으로 가감속 제어가 이루어졌음을 확인할 수 있다.

2.3.2. 삼각형 프로파일

짧은 거리 이동의 예외 상황에서 삼각형 프로파일 속도 명령 생성을 확인하고자 다른 조건은 고정한 채 최대 속도를 3000으로 증가시켰다.

$$S_1 = \frac{v_{max}^2}{2a} = \frac{3000^2}{2 \cdot 20000} = 225$$

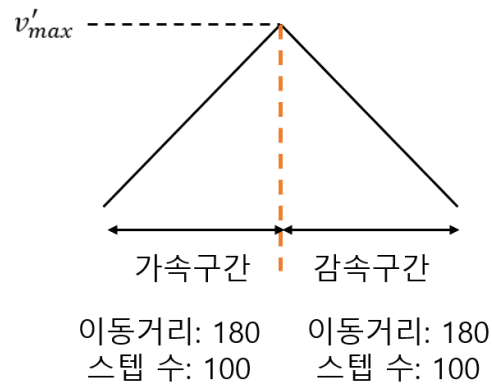
감속 구간에서 이동할 거리와 총 이동 거리 사이의 관계가 $S_{ref} \leq 2S_1$ 이므로, 등속 구간이 발생하지 않는다.

이 때, 수식적으로 계산한 각 구간별 이동 거리와 스텝 수는 다음과 같다.

$$\text{가속 구간 회전 각도 } S_1 = \frac{S_{ref}}{2} = 180 \quad \text{가속 구간 스텝 수 } k_1 = \frac{S_1}{\Delta\theta} = \frac{180}{1.8} \approx 100$$

$$\text{감속 구간 회전 각도 } S_{ref} - S_2 = 180 \quad \text{감속 구간 스텝 수 } k_2 = \frac{S_{ref} - S_2}{\Delta\theta} = \frac{180}{1.8} \approx 100$$

$$\text{최대 속도 } v'_{max} = \sqrt{a \cdot s_{ref}} = \sqrt{20000 \cdot 360} = 2683$$



[그림 9: 360deg, 3000 deg/s, 20000 deg/s²에서의 사다리꼴 속도 프로파일]

MACRO_PRINT 함수를 이용하여 각 구간 별 인터럽트 발생 횟수를 확인한 결과, 다음의 결과를 얻었다.

| | 가속구간 | | | | | | | 감속구간 | | | | | | |
|---------|------|-----|-----|-----|----|----|-----|------|-----|-----|-----|-----|-----|-----|
| step | 1 | 2 | 3 | ... | 98 | 99 | 100 | 101 | 102 | 103 | ... | 198 | 199 | 200 |
| 인터럽트 개수 | 670 | 474 | 387 | ... | 67 | 67 | 67 | 67 | 67 | 67 | ... | 387 | 474 | 670 |

- 100개 스텝의 가속 구간: 첫 구간 670개부터, 100번째 스텝의 67개까지 인터럽트 발생 횟수가 줄어들었다.
- 100개 스텝의 감속 구간: 101번째부터 200번째 스텝까지 인터럽트 발생 횟수가 증가하였다.

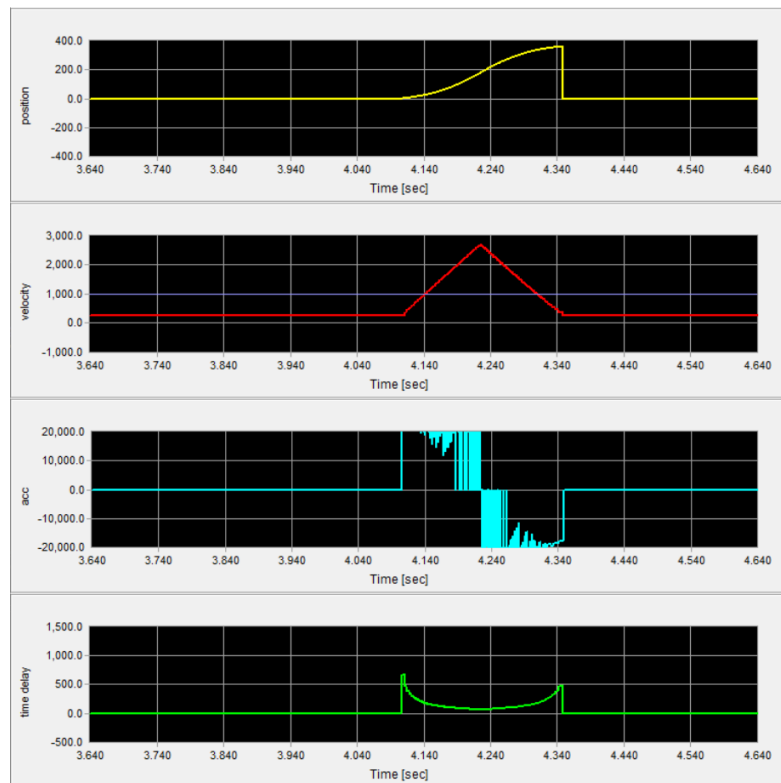
실험 결과, 최소 인터럽트 발생 개수는 90으로, 100,000 Hz 주기의 타이머 인터럽트를 사용할 때 속도는 다음과 같다.

$$v = \frac{\Delta\theta}{67/100000} = 2686$$

일부 오차가 발생하였지만, 인가한 속도 명령과 비슷한 최대 속도가 발생하는 것을 알 수 있으며, 수식으로 계산한 삼각형 속도 프로파일이 만들어지는 것을 확인할 수 있다.

USB 모니터로 파형을 출력한 결과는 다음과 같다.

Stepping Motor Control



[그림 10: 360deg, 3000 deg/s, 20000 deg/s²에서의 USB Monitor 출력 파형]

출력 파형은 차례대로 모터의 위치, 속도, 가속도, 각 스텝 간 상 지연 시간이며, 짧은 이동 거리 예외 상황에서 등속구간이 발생하지 않고, 정상적으로 삼각형 속도 프로파일에 해당하는 가감속 제어가 이루어졌음을 확인할 수 있다.

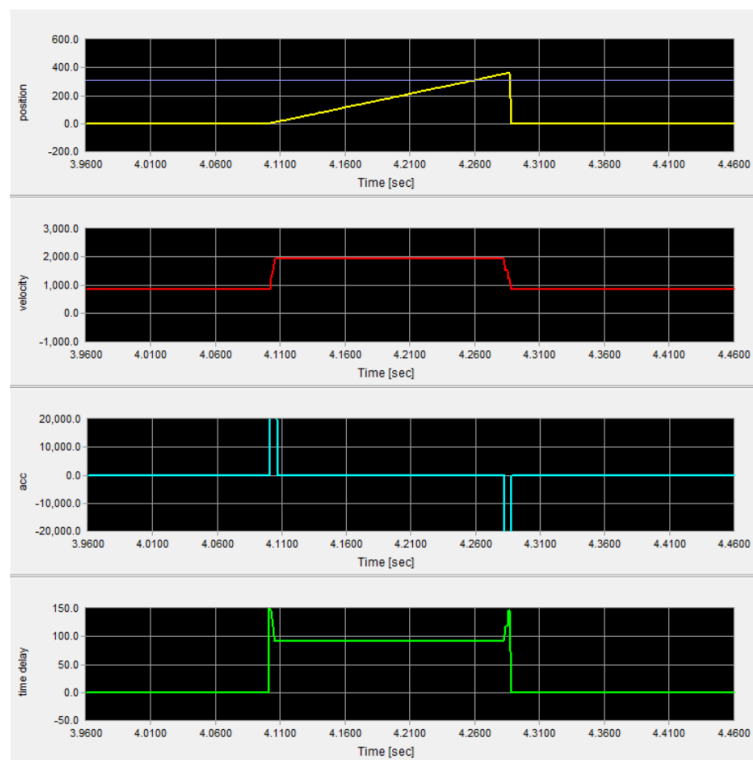
2.4. Stepping Motor 고속 제어를 위한 가속도 및 속도 튜닝

2.4.1. 가속도 튜닝

Stepping Motor 고속 제어를 위한 최대 가속도를 설정하고자 속도 및 위치 명령은 고정하고, 탈조가 발생할때까지 가속도를 점진적으로 증가시켜보았다.

그 결과, $300,000 \text{ deg/s}^2$ 에서 탈조가 발생하였고, 모터 구동의 안정성을 고려하여 최대 가속도의 70%인 $210,000 \text{ deg/s}^2$ 를 최종 가속도로 설정하였다.

이 때 USB 모니터를 통한 출력 파형은 다음과 같다.



[그림 11: 360deg, 2000 deg/s, $210,000 \text{ deg/s}^2$ 에서의 USB Monitor 출력 파형]

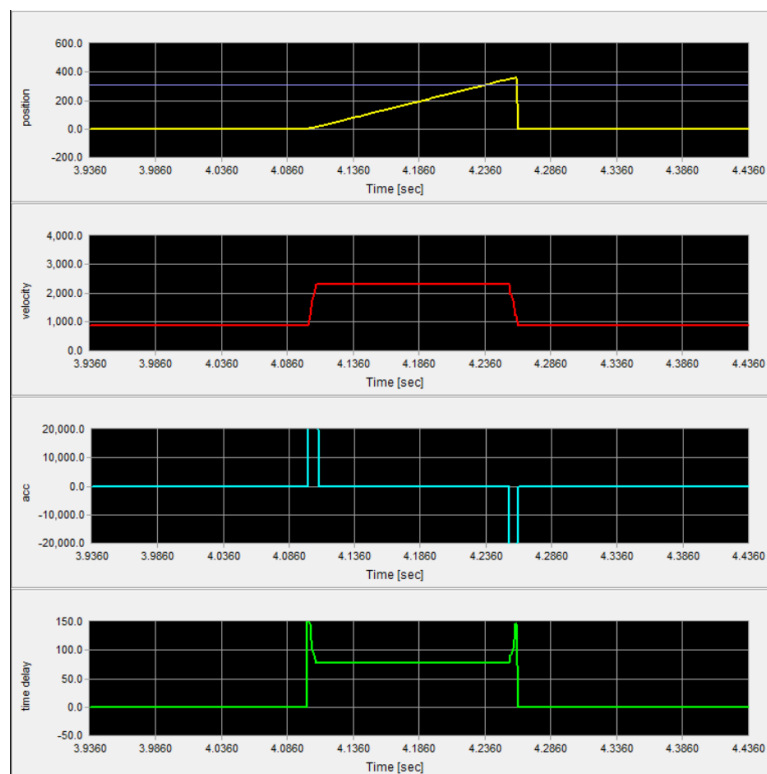
사다리꼴 속도 프로파일에서 가속구간 및 감속구간의 기울기가 가파른 것을 통해 큰 가속도의 고속 제어가 이루어졌음을 확인할 수 있다.

2.4.2. 속도 튜닝

Stepping Motor 고속 제어를 위한 최대 속도를 설정하고자 가속도 및 위치 명령은 고정하고, 탈조가 발생할때까지 속도를 점진적으로 증가시켜보았다. 이 때 가속도는 앞서 찾은 최대 가속도의 70%인 $210,000 \text{ deg/s}^2$ 를 사용하였다.

그 결과, 3500 deg/s 에서 탈조가 발생하였고, 모터 구동의 안정성을 고려하여 최대 속도의 70%인 2450 deg/s 를 최종 가속도로 설정하였다.

이 때 USB 모니터를 통한 출력 파형은 다음과 같다.



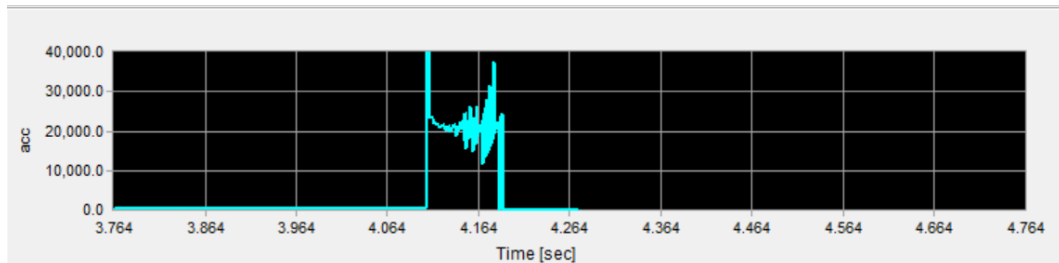
[그림 12: 360deg , 2450 deg/s , 210000 deg/s^2 에서의 USB Monitor 출력 파형]

사다리꼴 속도 프로파일에서 가속구간 및 감속구간의 기울기 및 최대 속도를 통해 큰 가속도 및 속도의 고속 제어가 이루어졌음을 확인할 수 있다.

3. 결론

3.1. 오차 분석

앞선 실험 과정에서, 속도가 빨라질수록 가속도 출력 파형에 오차가 커지는 것을 확인하였다.



[그림 13: 360deg, 2000 deg/s, 20000 deg/s²에서의 가속도 양상]

이는 인터럽트 발생 횟수의 정밀도와 관련하여 발생하는 오차이다.

인터럽트 발생 횟수는 다음과 같은 코드로 작성할 수 있다.

```
delayCnt = TIMER_FRQ * stepDelayTime; // TIMER_FRQ = 1e5
```

즉, 인터럽트 발생 횟수(delayCnt)는 상 전환 시간(stepDelayTime)에 실험에 사용되는 timer interrupt frequency인 100,000이 곱해진 후, 소수점 이하는 절사 되어 정수형으로 변환된다. 이 때 절사 되는 값은 상 전환 시간(stepDelayTime)이 작을수록 그 비중이 크기에, 속도가 빨라질수록 무시되는 값의 비중이 커져 큰 가속도 오차가 발생하는 파형이 출력되는 것이다.

3.2. 결과 고찰

실험 과정을 통해 DC 모터와 비교하여 Stepping motor는 정지마찰력에 의한 영향이 적으며, 오픈 루프 제어에서도 스텝 각 단위의 각도 정밀도가 보장된다는 것을 확인할 수 있었다. 또한, 탈조 현상을 유의하기 위한 가감속 제어 기법의 도입으로 고속 제어가 가능함을 확인할 수 있었다.