

Stepping Motor Control 예비보고서

20101449 한가은

1. 실험 개요

1.1. 실험 목적

1.2. 실험 대상 및 배경

1.2.1. Stepping motor란

1.2.2. Stepping motor에 사용되는 제어 기법

2. 실험 내용

2.1. 실험 절차

2.1.1. GPIO를 이용한 stepping 모터 구동

2.1.2. Stepping 모터 구동 API 작성

2.1.3. 속도 프로파일의 적용

3. 예상 결과

3.1. 실험 절차 2.1.1. GPIO를 이용한 stepping 모터 구동에서의 예상 결과

3.2. v_{\max} 고정, a 증가 튜닝에 따른 사다리꼴 속도 프로파일 변화

3.3. a 고정, v_{\max} 증가 튜닝에 따른 사다리꼴 속도 프로파일 변화

1. 실험 개요

1.1. 실험 목적

Stepping motor를 원하는 position 및 velocity가 되도록 제어한다. 사다리꼴 속도 프로파일을 통한 가감속 제어 기법을 통해 탈조가 발생하지 않는 최대 속도로 모터를 제어해본다.

1.2. 실험 대상 및 배경

1.2.1. Stepping motor란

스텝 각을 기본 회전 단위로 하는 모터로, 고정자로 코일을, 회전자로 영구자석을 갖는다. 전기회로에 의존하여 코일에 흐르는 전류 상을 전환하여 회전 방향 및 속도를 결정한다.

세 구동 방식, Full step, Half step, Micro step이 있으며, 각 구동 방식에 따른 분해능은 다음과 같다.

- Full step 구동: 분해능 = 스텝각
- Half step 구동: 분해능 = 스텝각/2
- Micro step 구동: 분해능 = 스텝각/N (N: 스텝 수)

Stepping motor 사용 시 코일 전류 상의 변화 속도를 따라가지 못해 고정자가 제자리에서 진동하는, 탈조 현상을 유의해야 한다.

1.2.2. Stepping motor에 사용되는 제어 기법

Stepping motor는 통상적으로 open loop으로 제어된다.

- overshoot을 줄이는 제어기 없음 → **micro step 구동을 통해 진동 감소**
- 탈조 현상 발생 여부 감지 불가 → **가감속 제어 기법 도입**

<Stepping motor의 위치 제어>

- 회전 각도 \propto 스텝 수
- 스텝 각 $\Delta\theta$, 상 전환 횟수 k , micro step 수 N 에 대해,
 - Full step: $s = \Delta\theta \times k$
 - Half step: $s = \frac{\Delta\theta}{2} \times k$
 - Micro step: $s = \frac{\Delta\theta}{N} \times k$

<Stepping motor의 속도 제어>

- 회전 속도 $\propto \Delta t$
- 스텝 각 $\Delta\theta$, 상 전환 시간 Δt , micro step 수 N 에 대해,
 - Full step: $v = \frac{\Delta\theta}{\Delta t}$
 - Half step: $v = \frac{\Delta\theta}{2 \times \Delta t}$
 - Micro step: $v = \frac{\Delta\theta}{N \times \Delta t}$

2. 실험 내용

2.1. 실험 절차

실험은 세 단계로 진행되며, 각 단계는 다음과 같다.

2.1.1. GPIO를 이용한 stepping 모터 구동

- Skeleton code를 통한 1상여자 구동방식을 사용
- delay time 변경을 통해 모터의 회전 속도 변경
- 과도하게 짧은 delay time을 설정하여 탈조 현상 확인
- Half step 구동 구현
- 모터 1회전에 필요한 스텝 계산
 - Full step: $\theta = 360/1.8 = 200steps$
 - Half step: $\theta = 360/(1.8/2) = 400steps$

2.1.2. Stepping 모터 구동 API 작성

스테핑 모터 구동을 위한 두 함수를 작성한다.

- `void OneStepMove(unsigned int dir, unsigned int tDelayCnt)`
 - 모터를 정방향 or 역방향으로 1 스텝 회전시킨다.
 - *dir*: 0은 정방향, 1은 역방향
 - *tDelayCnt*: 상 변환 후, *tDelayCnt*에 지정한 횟수 만큼 타이머 인터럽트가 발생할때까지 대기
 - 현재 인가된 상의 상태를 알아야 다음 상태를 생성할 수 있음 → global variable.
- `void OneStepMove(unsigned int dir, unsigned int tDelayCnt)`
 - 모터를 원하는 각도까지, 원하는 속도로 회전시킨다.
 - *angle*: angle에 지정한 스텝 만큼 회전. 양수면 정방향, 음수면 역방향.
 - *spd*: 회전 스피드, deg/s 단위.
 - *angle*과 *spd*로부터 회전 방향, 스텝 수, 스텝간 시간 지연을 타이머 인터럽트 주기 수로 환산하고, *OneStepMove*를 호출하여 구현한다.

StepMoveCV 함수 호출 시, *spd* 초기값을 너무 크게 설정하면 탈조가 발생하니 주의한다.

2.1.3. 속도 프로파일의 적용

감가속 제어를 위한 사다리꼴 속도 프로파일을 생성하기 위해 두 함수를 작성한다.

- `unsigned int MakeVelProfile(float maxVel, float accel)`
 - 가속 구간에 대해서만 '*maxVel*, *accel*' 두 개의 parameter를 사용하여 룩업테이블을 작성.
 - 룩업테이블은 200개를 넘지 않도록
- `void StepMoveVP(float angle, float maxVel, float accel)`
 - *MakeVelProfile* 함수를 호출 → 룩업테이블을 생성, 이를 활용해 스텝 모터 회전.
 - 사다리꼴 속도 프로파일을 결정하기 위한 3개의 parameter: '*angle*, *maxVel*, *accel*'. '*maxVel*, *accel*'은 절대값으로만 사용.
 - 가속 구간: 룩업테이블 index 증가
 - 등속 구간: *min_delay* 사용(일정한 index)
 - 감속 구간: index 감소

StepMoveVP 함수를 호출할 때, *maxVel*과 *accel* 값이 너무 크지 않게, *angle*값은 너무 작지 않게 설정할 것에 유의한다. 또한, 기구의 공진 주파수와 상전환 주파수가 일치해 진동으로 인해 탈조가 발생한 상황인지 확인한다.

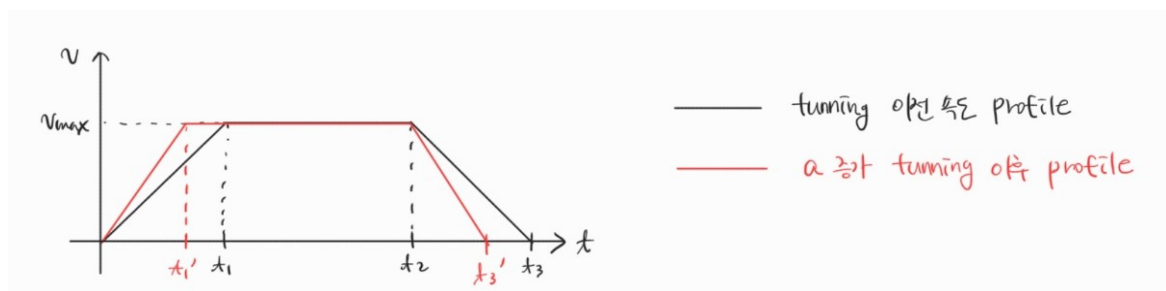
3. 예상 결과

3.1. 실험 절차 2.1.1. GPIO를 이용한 stepping 모터 구동에서의 예상 결과

- 모터 1회전에 필요한 스텝 계산
 - Full step: $\theta = 360/1.8 = 200steps$
 - Half step: $\theta = 360/(1.8/2) = 400steps$
- 모터 1회전 시 두 구동 방식에서의 속도 차이
 - Full step: $v = \frac{\Delta\theta}{\Delta t}$
 - Half step: $v = \frac{\Delta\theta}{2 \times \Delta t}$
 - 두 식으로부터, $\Delta\theta$ 및 Δt 가 일정할 때, Half step에서의 구동 속도가 Full step 대비 2배 느려짐을 예상할 수 있다.

3.2. v_{max} 고정, a 증가 튜닝에 따른 사다리꼴 속도 프로파일 변화

- v_{max} 가 일정한 상황에서 a 를 증가시켜 얻을 수 있는 변화된 사다리꼴 프로파일은 다음과 같다.



- 이는 다음 과정으로, 등가속도 운동 공식에 의해 각 구간에서의 변화를 예상할 수 있다.

i) 가속구간의 변화

$$\cdot t_1 = \frac{v_{max}}{a} \Rightarrow t_1 \text{ 감소}$$

$$\cdot s_1 = \frac{v_{max}^2}{2a} \Rightarrow s_1 \text{ 감소}$$

ii) 등속구간의 변화

$$\cdot \text{등속구간의 시간 } t = t_2 - t_1$$

$$\Rightarrow t_2 = \frac{s_{ref}}{v_{max}} \Rightarrow \text{변화 } X, t_1 \text{은 감소} \therefore t \text{등속은 증가}$$

$$\cdot \text{등속간 행진 거리 } s = s_2 - s_1 = (s_{ref} - s_1) - s_1 = s_{ref} - 2s_1$$

$$\Rightarrow s_1 \text{이 감소하므로 } s_{등속} \text{은 증가}$$

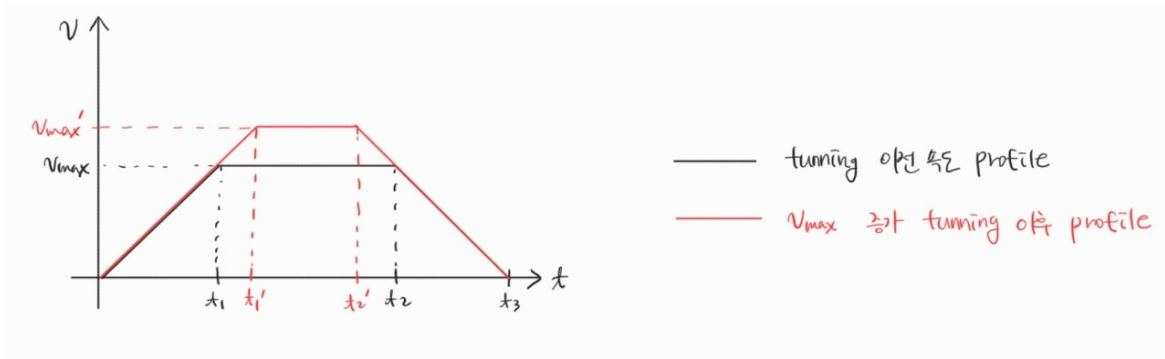
iii) 감속구간의 변화

$$\cdot t_3 = t_1 + t_2 \Rightarrow \text{감소}$$

- 모터 출력 토크에는 한계가 있으므로, a 값의 증가에 따라 탈조 현상을 확인할 수 있을 것이며, safety zone을 고려하여 a 값의 최대 속도를 설계해야 한다.

3.3. a 고정, v_{\max} 증가 튜닝에 따른 사다리꼴 속도 프로파일 변화

- a 가 일정한 상황에서 v_{\max} 를 증가시켜 얻을 수 있는 변화된 사다리꼴 프로파일은 다음과 같다.



- 이는 다음 과정으로, 등가속도 운동 공식에 의해 각 구간에서의 변화를 예상할 수 있다.

i) 가속구간의 변화

- $t_1 = \frac{v_{\max}}{a} \Rightarrow t_1 \text{ 증가}$
- $s_1 = \frac{v_{\max}^2}{2a} \Rightarrow s_1 \text{ 증가}$

ii) 등속구간의 변화

- 등속구간의 시간 $t_{\text{등속}} = t_2 - t_1$
 $\Rightarrow t_2 = \frac{s_{\text{ref}}}{v_{\max}} \Rightarrow \text{감소}, t_1 \text{은 증가. } \therefore t_{\text{등속}} \text{은 감소.}$
- 등속간 움직인 거리 $s_{\text{등속}} = s_2 - s_1 = (s_{\text{ref}} - s_1) - s_1 = s_{\text{ref}} - 2s_1$
 $\Rightarrow s_1 \text{이 증가하므로 } s_{\text{등속}} \text{은 감소.}$

iii) 감속구간의 변화

- $t_3 = t_1 + t_2 \Rightarrow \text{동일}$

- 모터 출력 최대 속도에는 한계가 있으므로, v_{\max} 값의 증가에 따라 탈조 현상을 확인할 수 있을 것이며, safety zone을 고려하여 v_{\max} 값의 최대 속도를 설계해야 한다.