

## 저속영역의 속도검출특성 개선을 위한 새로운 속도검출방법

백상기<sup>○</sup> 민종진 이진우  
 삼성종합기술원, System & Control Sector

### New Speed Detection Method for the Improvement of the Speed Detection Characteristics in the Low Speed Region

S.K. Baek J.J. Min J.W. Lee  
 System & Control Sector, Samsung Advanced Institute of Technology

#### Abstract

A new speed detection method using low resolution incremental encoders, especially excellent in the low speed region and in the transient state, is proposed. The half period error of an incremental encoder is greater than the period error. So it's not recommended to quadruple the number of pulses per revolution, because it increase the ripple of speed. To overcome this restriction a speed detection method has been proposed. But it requires so many latch circuits. Therefore we propose a new speed detection method that has different concept and has fewer latch circuits.

#### 1. 서론

고성능을 필요로 하는 산업분야에 적용되고 있는 유도전동기 벡터제어시스템은 속도검출 장치로 대부분 증분형(incremental)엔코더를 사용하고 있다. 벡터제어에 사용되는 회전자의 위치를 얻는 목적도 있지만 주 목적은 속도제어에 사용될 속도정보를 얻는 것이다. 증분형엔코더는 90°의 위상차를 갖는 두개의 펄스열을 출력하며 펄스열의 주파수는 모터속도에 비례한다. 그러므로 일정시간 동안의 펄스갯수를 안다면 속도검출을 할 수 있다. 반대로 한 펄스의 시간을 알아도 속도검출이 가능함을 알 수 있다. 칠판계 방법은 저속에서 오차가 많으며 두번째 방법은 고속에서 오차가 많으므로 이 두방법을 혼용하여 사용하는 방법이 보편적으로 사용되고 있다[1]-[3].

일반적으로 쓰이는 증분형엔코더는 펄스의 주기오차에 비해 반 주기오차가 크기 때문에 4채배된 신호로 속도검출을 하게 되면 반 주기오차에 비례하는 속도검출 오차가 생기게 된다. 그러므로 펄스주기의 정수배의 각도차를 이용해 속도검출하는 것이 속도리플 저감에 유리하다. 그러나 저속영역으로 갈수록 엔코더 펄스의 주기는 속도검출주기에 비해 커지게 되고 이에 따른 속도검출지연도 커지게 되어 시스템의 불안정 요인이 되므로 이때는 4채배 속도검출방법으로 전환하는 방법도 쓰이고 있다.

이와 같은 속도검출 상의 제한요인을 없애기 위해 엔코더 펄스의 4채배된 신호를 이용하면서 두 상간의 위상오차는 포함하지 않는 새로운 속도검출방법이 발표되었는데, 이 방법의 특징은 속도검출구간을 겹치게 하여 엔코더 펄스의 모든 에지(edge)정보를 사용하면서 정수배의 펄스주기를 속도검출에 이용하는 것이다[4],[5].

이 방법은 엔코더 펄스의 모든 에지에 대한 시간 및 각도정보를 계속 래치(latch)해 가면서 속도검출 주기마다 또 다른 래치로 테

이터를 저장시키는 구조로 되어 있어 많은 양의 래치회로를 필요로 한다.

본 논문에서는 속도검출 성능을 유지하면서 래치회로를 대폭 줄인 새로운 개념의 속도검출방법을 제안한다. 제안된 속도검출방법의 개념 및 하드웨어 구성을 보이고, TMS320C31을 사용한 알고리즘 수행결과를 보인다.

#### 2. 기존 속도검출방법의 문제점

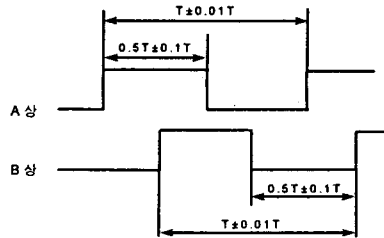


그림 1. 증분형 엔코더의 출력펄스 파형 예

현재 전동기의 속도검출에 널리 쓰이고 있는 증분형 엔코더는 그림 1과 같이 90° 위상차를 갖는 A, B 두상의 펄스열을 출력하며, 속도검출은 일정시간 dt 동안 변화한 각도 dθ를 이용하여  $\omega = d\theta/dt$ 로 구할 수 있다. 엔코더 출력펄스는 그림 1의 예와 같이 가공상의 오차를 갖는다. A상 또는 B상 펄스의 반주기는 10%의 오차를 갖는데 반해 주기는 1%의 오차만을 나타내므로, 속도검출에는 체배 신호 대신 A상 또는 B상 펄스의 정수배 신호만을 사용하는 것이 속도검출의 정확도 면에서 유리하다. 이 경우 한 상의 신호중 상승에지 또는 하강에지만이 속도검출에 사용되므로 전체 엔코더정보의 25%만이 속도검출에 사용되게 된다. 그림 2는 이와 같은 제한요인을 가진 일반적인 속도검출 방법을 나타낸다. 속도검출이 이루어지기 위해서는 속도검출주기 내에 두개 이상의 상승에지가 발생되어야 하며 그렇지 않을 경우 검출지연으로 인한 시스템 불안정이 발생한다. 저속추정용 속도관측기(speed observer)를 병행하여 사용한다고 해도 극저속에서는 추정기간이 길어져 추정오차가 커지게 된다. 또한 속도검출주기 간에 데이터를 읽는 시간이 필요해 속도검출 제시각이 바로 연결이 안되는 문제점도 있는데 이 문제를 해결하기 위해서는 하드웨어 보강이 필요하다. 이러한 단점에도 불구하고 이 방법이 널리 쓰이는 이유는 구조가 간단하여 구현이 용이하며 중, 고속에서는 만족할 만한 속도검출이 이루어지기 때문이다.

이와 같은 방법의 속도검출 방법에 속도관측기를 병행하여 사용할 경우 별 무리없이 사용할 수도 있지만 좀 더 정확한 저속영역

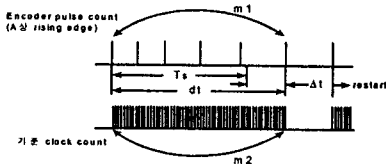


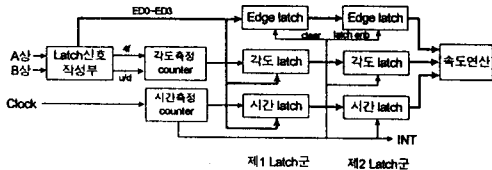
그림 2. 일반적인 속도검출 방법

의 제어성능을 얻기 위해서는 엔코더 정보를 100% 사용하여 최대한 정확한 속도검출을 할 필요가 있다.

### 3. 엔코더정보를 100% 사용하는 속도검출 방법

기존 속도검출방법의 한계를 해결하기 위해서는 엔코더펄스의 모든 에지정보를 이용하면서 반주기 오차의 영향을 받지 않도록 해야 한다. 반주기 오차의 영향을 받지 않기 위해서는 정수배의 엔코더펄스로 속도연산을 행해야 함을 알 수 있는데 이를 위해 속도검출구간을 겹치게 하는 방법이 개발되었으며[4][5], 이의 구성 및 방법을 그림 3과 그림 4에 나타내었다.

각도측정 카운터는 엔코더 펄스의 에지가 발생할 때마다 up 또는 down count 되며, 시간측정 카운터는 기준클럭을 up count 하여 Ts (Sampling Time)의 주기마다 reset 된다. 제1 래치군은 엔코더펄스의 각 에지가 발생할 때마다 그 시점의 에지정보, 각도측정 값 및 시간측정 값을 저장하며, 제2 래치군은 Ts의 주기마다 제1 래치군의 데이터들을 저장한다. 각도래치 및 시간래치들은 A상 상승/하강에 지, B상 상승/하강에 지 데이터들을 저장하기 위해 각각 4개의 래치로 구성된다. 현재의 속도검출시점을 기준으로 할 때 제2 래치군에 각 에지에 해당하는 최신의 각도 및 시간 데이터가 저장되며, 이전의 데이터들은 연산처리장치에 저장되어져 있게 된다. 제2 래치군에 저장된 데이터들은 연산처리장치에 의해 임혀져 이전의 데이터와 함께 속도계산에 사용된다. 속도는 그림 4와 같이 Ts 주기내의 최종에지와 그 에지와 같은 종류인 이전 에지의 시간 및 각도 데이터로부터 구해진다. 이 방법은 시간 및 각도데이터를 저장하는 하드웨어를 이종으로 하여 엔코더 정보를 100% 이용할 수 있도록 하였으며, 속도검출주기 동안 에지가 한개만 발생되어도 이전 데이터를 이용하여 속도검출이 가능하게 된다. 그러나 이 방법은 그림 3에서 보는 바와 같이 상당히 많은 래치회로를 필요로 하게 되어 구현시 하드웨어 부담을 갖게 한다.



- 각도측정 counter 10bits, 시간측정 counter 14bits인 경우
- 제1 latch군: 100bits (Edge latch 4bits, 각도 latch 10bits×4=40bits, 시간 latch 14bits×4=56bits)
- 제2 latch군: 100bits
- ⇒ 총 200bits latch 필요

그림 3. 엔코더정보를 100% 사용하기 위한 속도검출기의 구성

### 4. 새로운 속도검출 방법

그림 3 에서 제1 래치군은 속도검출 주기에 관계없이 계속 래치 동작을 수행해야 하므로 제1 래치군의 래치동작을 멈추지 않고 매 속도검출주기 마다 속도연산에 필요한 데이터를 얻기 위해서는 제1

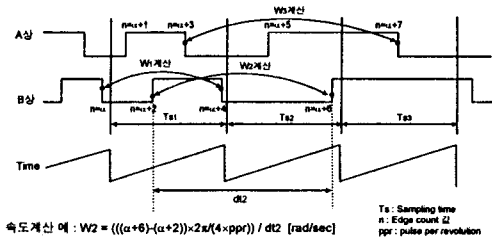
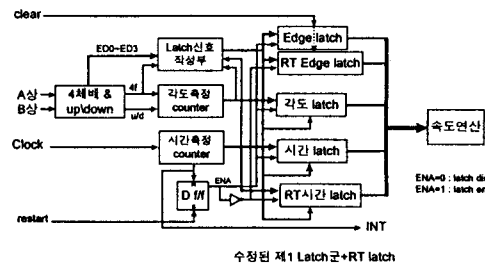


그림 4. 그림 3구성의 속도계산 방법

래치군과 같은 구성을 갖는 제2 래치군이 필요하게 된다. 단지 속도연산에 필요한 데이터를 잠시 저장하기 위해 또 다른 래치군을 두는 것은 하드웨어 낭비라 할 수 있다. 또한 속도연산에 필요한 각도 정보는 한주기내에 발생된 총 에지의 갯수만으로도 알 수 있으므로 에지 종류별로 각도래치를 하나씩 두는 것도 역시 하드웨어 낭비이다. 본 논문에서는 이와 같은 하드웨어 낭비요소를 없앤 새로운 개념의 속도검출방법을 그림 5와 같이 구성하였다.



- 각도측정 counter 10bits, 시간측정 counter 14bits인 경우
- Edge latch : 4bits
- RT Edge latch : 4bits
- 각도 latch : 10bits
- 시간 latch : 14bits × 4 = 56bits
- RT시간 latch : 14bits
- ⇒ 총 88bits latch 필요

그림 5. 새로운 속도검출기의 구성

기본구성은 엔코더 정보를 100% 이용하면서 하드웨어를 최소화하는 구조로, 그림 3의 제2 래치군을 없애고 제1 래치군의 각도래치도 각 에지별로 한개의 래치로 구성되어 있던 것을 1/4로 줄여 에지의 종류에 상관없이 현재의 각도측정카운터의 값을 래치하도록 하였다. 제2 래치군을 없애는 대신 제1 래치군에 래치된 각도 및 시간 데이터를 읽는 동안 발생하는 에지에 대한 정보 및 시간 데이터를 래치하기 위한 RT(Read Time) 래치를 구성하였다. 이와 같은 구성으로 래치회로를 50%이하로 줄일 수 있음을 알 수 있다.

제1 래치군과 RT 래치사이의 래치 제어신호는 D/f 출력인 ENA 신호로, 속도검출주기가 되어 INT신호가 발생이 되면(ENA=0) 제1 래치군은 래치동작을 멈추고 최종의 값을 유지하게 되고 이때부터 RT 래치는 enable이 되어 연산장치가 데이터를 읽는 동안 발생하는 엔코더펄스 에지의 정보 및 시간 데이터를 저장하게 된다. 제1 래치군의 데이터들을 모두 읽고 에지래치를 clear한후 restart신호가 주어지면(ENA=1) 이때부터 제1 래치군의 래치동작은 다시 시작되고 RT래치는 disable 되어 최종의 값을 유지하고 있게 된다. 이 값들은 해당 속도검출주기의 속도연산이 끝난후 임혀져 다음 속도검출주기에 사용될 수 있도록 일정변수에 저장된다.

그림 6는 속도계산 예를 보인다. Ts2 구간에서 A상 상승에 지 및 B상 상승에 지가 발생했으므로 에지래치에는 이 에지들에 해당하는 bit만 "1"로 set된다. 에지래치로부터 이 정보를 얻고 A,B상 상승에 지에 해당하는 시간래치의 값을 비교해 보면 최종에지가 B상 상승에 지임을 알 수 있다. 저장되어 있는 이전의 B상 상승에 지의 시간

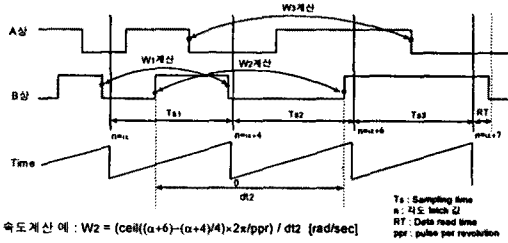


그림 6. 새로운 속도검출기의 속도계산 방법

데이터로부터  $dt2$ 를 계산할 수 있다. 각도래치의 값은 " $a+6$ "의 값을 가지고 있으므로 저장되어 있는 이전의 각도래치 값 " $a+4$ "와의 차로 부터 발생된 에지갯수는 2개임을 알 수 있다. 실제 속도계산은 펄스의 정수배로 계산이 되므로 에지가 2개 발생했다면 속도계산시 사용되는 펄스갯수는 1개임을 알 수 있다. 이와 같은 변환을 위해 ceil함수를 이용한다.  $Ts$  구간에서 제1레치군에 래치되는 에지정보가 없다면 RT래치의 정보인 B상 하강에지의 데이터가 속도계산에 사용된다.  $Ts$ 주기내에 발생된 에지의 시간데이터들은 속도계산후 모두 새로운 이전 값으로 갱신되어 이후의 속도계산에 사용된다. 이상의 속도계산 방법과 초기화 및 에지가 발생하지 않을 경우의 처리방법을 포함한 전체 속도계산 알고리즘을 그림 7에 나타내었다.

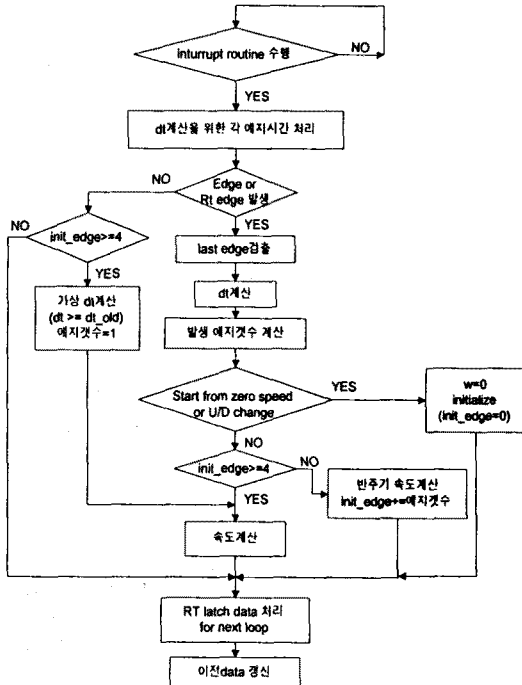


그림 7. 속도계산 알고리즘

### 5. 실험결과

EPLD로 하드웨어를 구성하고 TMS320C31을 사용하여 기존방법과 제안한 방법의 속도검출 특성을 비교하였다. 기존의 한 상만을 이용하는 속도검출방법과 새로운 속도검출방법과의 비교파형을 그림 8(a) 및 그림 9(a)에 나타내었으며, 50,000ppr인 정밀 엔코더와의 비교파형을 그림 8(b) 및 그림 9(b)에 나타내었다. 기존방법에

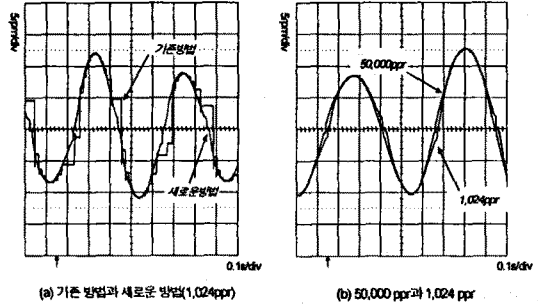


그림 8. 속도검출 실험결과 파형(5rpm/div)

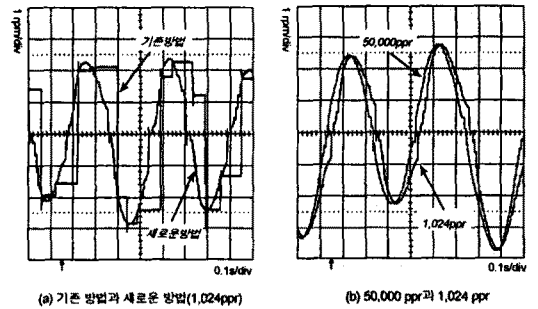


그림 9. 속도검출 실험결과 파형(1rpm/div)

비해 상당히 정밀하게 속도검출이 되는 것을 볼 수 있으며, 저분해능 엔코더의 한계로 인해 50,000ppr 엔코더와 비교할때는 delay가 약간 생기는 것을 볼 수 있다.

### 6. 결론

본 논문에서는 하드웨어를 대폭 줄이면서 엔코더 정보를 100% 모두 이용하는 새로운 속도검출 방법을 제시하였다. 실험결과에서 볼 수 있듯이 극 저속에서도 만족할 만한 속도검출이 이루어졌으며, 저분해능 엔코더의 한계로 인한 검출 delay는 다소 생기는 것을 볼 수 있었다. 이러한 한계점은 관측기 등을 통해 보상에 줄 수 있으며 이 보상이 적절히 되는 경우 극 저속에서도 상당히 정확한 속도검출 성능을 기대할 수 있다.

### 참고문헌

- [1] F. Briz, J.A. Cancelas, and A. Diez, "Speed Measurement Using Rotary Encoders for High Performance ac Drives", pp.538-542, IECON'94
- [2] Kouetsu Fujita and Katsumasa Sado, "Instantaneous Speed Detection with Parameter Identification for ac Servo Systems", IEEE Trans. Ind. Applicat., vol.28, no.4, pp.864-872, 1992
- [3] I Miyashita and Y Ohmori, "A New Speed Observer for an Induction Motor using the Speed Estimation Technique", pp.349-353., EPE'93
- [4] Yasuhiro Yamamoto, Yasuhiro Yoshida, and Tetsuo Yamada "極低速域の特性を改善する 速度検出方式と速度オブザーバへの適用", 平成6年電學産業應用全大, No.59
- [5] "速度検出装置", 日本特許廳 公開特許公報 平6-118090