

Software PWM을 이용한 AC Servo Motor 제어기의 구현

○ 홍기철 남광희
포항공과대학 전자전기공학과

AC Servo Motor Control Using Software PWM

○ Kichul Hong, Kwanghee Nam
Dept. of Electrical & Electronic Engin. POSTECH

Abstract - We utilize as a processor TMS320C25 (Texas Instrument) in making a driver for a 4 pole PM synchronous servo motor. TMS320C25 has a 32bit ALU and a 16 bit hardware multiplier, and the maximum instruction execution rate is 10MIPS at 40MHz. We adopted a space vector modulation PWM method. An interesting point of this work is that PWM wave is generated by utilizing timer interrupts. Hence, in the rest of time the processor can take care of the other routine such as Park's coordinate transformation and the computation required in the feedback loops. Thus, it makes the hardware circuit very simple. Due to the decrease in the number of components, the motor drive system becomes more fault-tolerant and cost-optimized. Also, more flexibility is gained in changing the control parameters.

I. 서 론

영구자석동기기의 구조를 가진 AC Servo Motor는 보수유지의 편리성 및 우수한 토크특성 때문에 서보시스템에서의 채택이 두드러지고 있다. 그러나, 제어기구성이 어렵고, 여러가지 장치등으로 인하여 회로의 구성이 많아지며, 이에 따라 고가가 되는 단점이 있다. 최근에는 이러한 문제들을 해결하기 위해서 마이크로프로세서를 이용하여 제어시스템을 디지털화 소형화하려는 연구가 활발히 진행중에 있다. 특히 고속프로세서인 DSP(Digital Signal Processor)를 사용하여 PWM을 software algorithm으로 처리함으로써 전체제어기 구성을 완전 디지털화 하려는 노력이 경주되고 있다. AC Servo를 위한 완전디지털제어기(full digital controller)로 구성하기위해 본 연구에서는 고속 DSP인 TMS320C25(Texas Instrument사)를 채택하였다. 특히 PWM을 프로세서 내부에서 처리하기위해서 TMS320C25에 내장된 timer interrupt structure를 사용하여 처리함으로써 전체 제어시스템이 소형화 되었다.

II. 본 론

그림 1은 전체제어기의 블럭도이다. 측정된 상 전류는 ADC(Analog-to-digital converter)를 통하여 프로세서에 입력되며 이 값은 좌표변환을 통하여 전류제어기에 입력된다. 내부에 전류제어기가 들어 있고 그 밖에 속도제어기가 구성된 형태이다. 속도제어와 전류제어기가 모두 프로세서내부에서 계산으로 처리되며, PWM 파형발생기도 프로세서의 timer interrupt를 이용하여 처

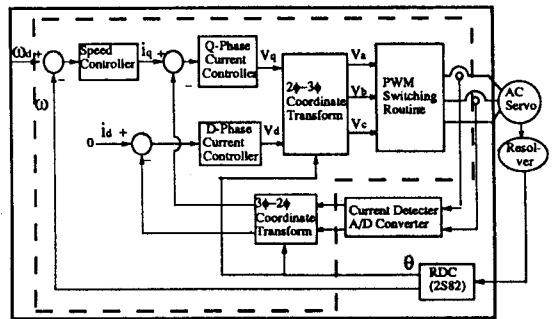


그림 1 AC Servo 제어기 블럭도

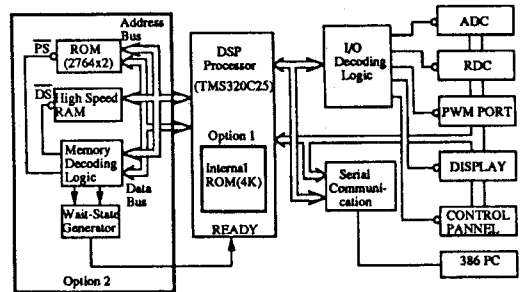


그림 2 TMS320C25 Main Control Board 블럭도

리된다. 상 전류 검출을 위해서는 Hall sensor를 이용한 상용화된 전류검출기를 사용하였다. 여기서 출력된 전압은 12 bit의 해상도를 갖는 ADC(Analog-to-digital converter) chip을 통하여 디지털값으로 변환되어, data bus를 통하여 프로세서에 입력된다. 위치 및 속도검출을 위해서 AC servo에 부착된 레졸바블(Resolver)를 사용하였다. 레졸바에서 절대 위치정보를 얻기위해의 RDC(Resolver Digital Converter) chip인 2S82(Analog Devices사)를 사용하였으며 14 bit/rev의 위치해상도를 갖도록 구성되었다.

그림 2는 TMS320C25를 중심으로 하여 구성된 Main Control Board의 구성도이다. Option 2에 해당하는 외부 ROM과 RAM을 부착할 경우는 이 외부 메모리를 위한 decoding logic이 필요하며, low-speed memory(Access time > 40 ns)를 사용할 경우는

wait-state generator logic 이 필요하게 된다. 그러나 상용화할 경우 option 1와 같은 ROM이 내장된 TMS chip을 선택하면 option 2의 외부메모리 및 이를 위한 회로구성이 필요없게 되므로 전체 시스템의 구성이 대단히 간단해진다. 그외에 외부에서 검출된 전류 및 위치정보는 I/O bus를 이용하여 읽어들이며, 또한 프로세서나 AC Servo의 상태를 모니터링하기 위해서 LCD display기능을 추가하였다. 또한 상위기종, 특히 PC와의 통신을 위해서 serial communication logic 이 구성되어 있다.

III. Software PWM using Timer Interrupt

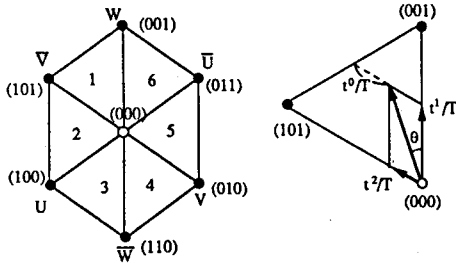


그림 3 Space Vector Representation Method

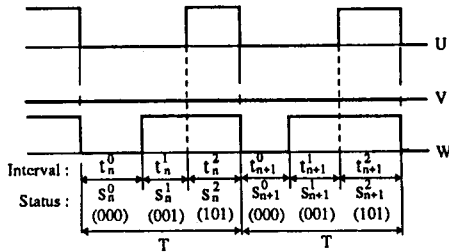


그림 4 공간벡터를 이용한 Timing Sequence 예제

PWM 파형발생 방법으로는 그림 3의 공간벡터를 이용한 변조방법을 사용한다. 이 방법을 이용하여 각 스위칭 상태의 duty 를 구하는 식은 다음과 같다.

$$\begin{aligned} t^1 &= qT(2/\sqrt{3}) \sin(\pi/3 - \theta) \\ t^2 &= qT(2/\sqrt{3}) \sin \theta \\ t^3 &= T - t^1 - t^2 \end{aligned} \quad (1)$$

여기서 q는 modulation factor 이다.

그림 4와 같은 공간벡터를 이용한 PWM 파형을 발생시키기 위해서 TMS320C25에 내장된 timer interrupt logic을 사용하였다. TMS320C25에는 TIM 레지스터와 PRD 레지스터라는 timer interrupt를 위한 두 개의 레지스터를 제공한다. TIM 레지스터는 CLKOUT1(10MHz)에 동기된 synchronous down counter로서 값이 0 이되면 timer interrupt를 발생시킨다. 그리고 그 다음 싸이클(cycle)에서 PRD 레지스터의 값을 복사한다. 따라서 새로운 timer interrupt가 발생하기 이전에 PRD에 새로운 값을 써넣으므로 원하는 시간간격을 갖는 interrupt를 발생시킬 수 있다. 따라서 식(1)에서 구한 t^i 를 매 interrupt 마다 교체함으로써 그림 4와 같은 시간간격을 갖는 파형을 발생시킬 수 있다.

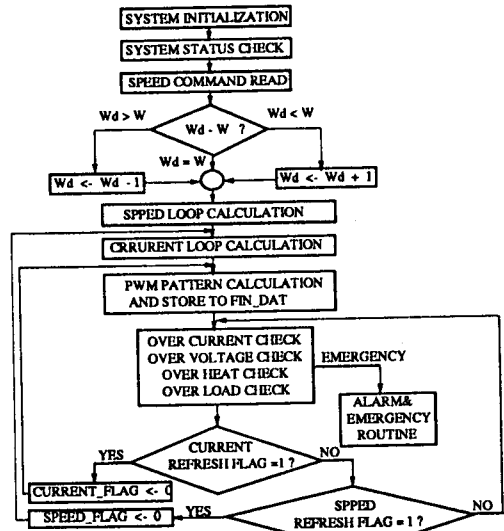


그림 5 Main Routine Flowchart

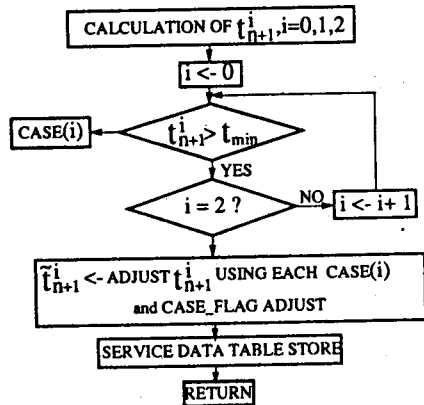


그림 6 PWM Pattern Adjustment Flowchart

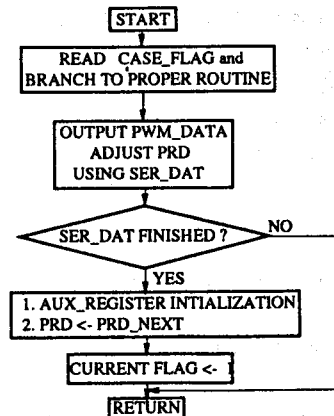


그림 7 Interrupt Service Routine Flowchart

그림 5는 main routine의 개략적인 흐름도이다. 전류제어 루프와 속도제어루프는 current refresh flag과 speed refresh flag을 이용하여 feedback되며 이 flag들의 refresh rate을 결정함으로써 속도제어기와 전류제어기의 sampling 주파수를 조정할 수 있도록 구성되어 있다.

Main routine에서 계산된 PWM 패턴은 ISR(interrupt service routine)이 사용할 수 있도록 그림 6의 흐름도와 같은 조정과정을 거치게 된다. 그림 4에서 t_1 가 t_{min} (interrupt service를 수행하는데 필요한 최소시간) 보다 작은 경우에는 ISR이 끝나기도 전에 새로운 파형이 발생되어야 하므로 정상적인 파형출력에 문제가 발생한다. 이러한 경우가 발생하지 않기 위해서 그림 7에서처럼 각 t_1 를 조사한 다음 $t_1 < t_{min}$ 인 경우에는 CASE_FLAG라는 메모리에 그 정보를 기록하여 ISR에서 적절한 처리가 가능하도록 조정한다. 예를 들어서 $t_1 < t_{min}$ 인 경우, S_1 을 출력하고 나서 t_1 동안 NOP(No operation) 명령을 이용하여 공전(idling)시킨다음, S_2 를 출력하고 복귀 한다. 이러한 스위칭간격의 조정을 ISR에서 처리할 수도 있으나 보다 효율적인 방법은 그림 6과 같이 각 시간간격의 상태를 조사하고 거기에 대한 정보를 미리 기록해 두는 방법이다. 이렇게 하면 ISR은 CASE_FLAG의 값만 보고 해당 시간간격에 대한 적절한 service를 수행할 수 있게 된다.

그림 7은 ISR을 수행하는 흐름도이다. 그림 6에서 조정된 스위칭간격 및 CASE_FLAG의 값을 보고 적합한 서비스를 수행하도록 구성되어 있다. 여기서 SER_DAT는 ISR이 사용하는 PWM 패턴이다.

IV. 실험 결과

그림 8은 공간벡터변조방법으로 발생된 U상과 V상의 스위칭파형이다. 그리고 그림 9는 그림 8에서 출력된 두개의 상 전압파형을 모터의 시정수와 일치하는 R-C필터에 통과시켜서 만든 sin파형이다. 스위칭주파수는 25KHz이며 sin파의 주파수는 100Hz이다. 그리고 그림 10은 제작된 AC Servo 제어기의 Prototype의 모습이다.

V. 결론

TMS320C25를 사용하여 AC Servo Motor 제어기를 구성하였다. PWM을 software로 처리하였으며 PWM을 위한 계산시간이 외에 남은 시간에 전류 및 속도제어에 필요한 연산을 모두 처리함으로써 완전디지털 제어기를 구성하였다. 따라서 전체 하드웨어의 구성이 간략해지며 이에 따른 부품감소로 제어기가 fault tolerant해지며 경비절감효과가 있다.

참고 문헌

- [1] Takashi Kenjo, Power Electronics for the Microprocessor Age, Oxford University Press, 1990.
- [2] Paul C. Krase, Analysis of Electric machinery, McGraw-Hill Book Company, 1987.
- [3] TMS32025 User/Guide, TI, 1990.

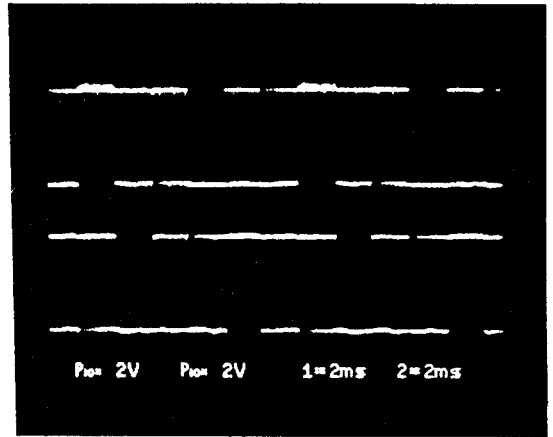


그림 8 공간벡터변조방법을 사용한 U상, V상 전압 파형

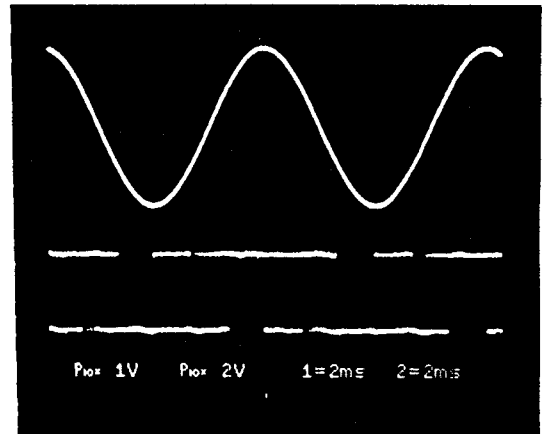


그림 9 선간전압파형을 R-C filtering 한 후의 sin파형

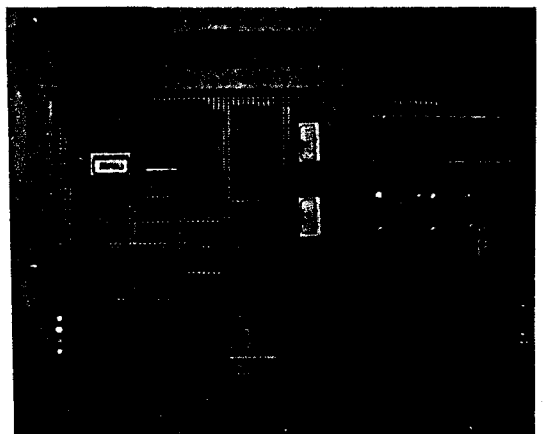


그림 10 AC Servo 제어기의 Main Controller Board