

고급 모터 제어 기법 구현 에너지 효율적인 모터 제어를 위한 센서리스 FOC

센서리스 FOC(Field Oriented Control)가 내장된 PMSM(영구 자석 동기 모터) 기반 고급 모터 제어 시스템의 채택을 가속화하는 핵심 원동력에는 두 가지가 있다. 바로 에너지 효율 개선과 제품 차별화 강화다.

글/넬슨 알렉산더(Nelson Alexander) 마이크로칩테크놀로지 수석 마케팅 엔지니어

센서리스 FOC(Field Oriented Control)가 내장된 PMSM(영구 자석 동기 모터) 기반 고급 모터 제어 시스템의 채택을 가속화하는 핵심 원동력에는 두 가지가 있다. 바로 에너지 효율 개선과 제품 차별화 강화다. 센서리스 FOC가 적용된 PMSM은 이 두 목표를 모두 달성할 수 있는 것으로 입증되었지만, 성공을 위해서는 전체론적인(holistic) 접근 구현 방식을 제공하는 디자인 에코시스템이 필요하다. 전체를 아우르는 에코시스템을 통해 개발자는 그동안 PMSM의 채택을 가로막은 구현 과제를 극복할 수 있다.

왜 PMSM인가?

PMSM 모터는 전자식 정류(commutation) 기능을 사용하는 브러시리스 모터이다. 이를 브러시리스 직류 모터(BLDC)와 혼동하는 경우가 많은데, BLDC도 브러시리스 모터 제품군에 속하며 전자식 정류를 사용하지만 구성 방식이 약간 다르다. PMSM의 구성은 FOC에 최적화되어 있지만 BLDC 모터는 6스텝 정류 기법 사용에 최적화되어 있다. 최적화 결과 PMSM은 정현파(sinusoidal)

역기전력(Back-EMF)이 생기고, BLDC 모터는 사다리꼴(trapezoidal) 역기전력이 생긴다.

각 모터와 함께 사용되는 회전자(rotor) 위치 센서에도 차이가 있다. PMSM은 대개 위치 인코더를 사용하지만 BLDC 모터는 동작을 위해 세 개의 홀(Hall) 센서를 사용한다. 비용이 부담이 된다면 자석, 센서, 커넥터 및 배선 비용이 들지 않는 센서리스 기법 구현을 고려해 볼 수 있다. 센서를 제거하면 시스템에서 고장 발생 가능성이 있는 부품이 줄어들기 때문에 제품의 신뢰성도 높아진다. 센서리스 PMSM과 센서리스 BLDC를 비교해 보면, FOC 알고리즘이 적용된 센서리스 PMSM은 비슷한 구현 비용으로 유사한 하드웨어 설계를 사용하면서 더 나은 성능을 확보할 수 있다.

PMSM 전환으로 가장 큰 혜택을 누리는 애플리케이션은 현재 브러시 DC(BDC)나 AC 인덕션 모터(ACIM)를 사용하고 있는 애플리케이션이다. PMSM으로 전환할 때 얻을 수 있는 주요 이점에는 전력 소비 감소, 속도 증가, 더 부드러운 토크, 가청 소음 저감, 수명 개선, 크기 축소, 애플리케이션의 경쟁력 강화 등이 있다. 그러나 이와 같은 PMSM 사용 효과를 실현하기 위해서는 시스템 요건에 맞

는 애플리케이션별 알고리즘과 더 복잡한 FOC 제어 기법을 구현해야 한다. PMSM은 BDC나 ACIM에 비해 더 높은 비용을 수반하지만 훨씬 더 많은 이점을 제공한다.

구현 과제

그러나 PMSM 사용의 이점을 실현하기 위해서는 고급 FOC 모터 제어 기법 구현에 내재된 하드웨어 복잡성과 그에 필요한 전문 지식을 이해해야 한다. **그림 1**은 3상 전압원 인버터를 적용한 3상 센서리스 PMSM 제어 시스템이다. 인버터를 제어하기 위해서는 상호 연결된 3쌍의 고해상도를 가진 PWM(펄스 폭 변조) 신호와 신호 제어에 필요한 아날로그 피드백 신호가 많이 필요하다. 또한 시스템에는 신속한 응답을 위해 고속 아날로그 비교기(comparator)를 사용하여 설계됨으로써 내결함성(fault tolerance)을 제공하는 하드웨어 보호 기능이 필요하다. 센싱, 제어 및 보호를 위한 이들 추가 아날로그 구성 요소는 일반적인 BDC 모터 설계나 ACIM의 간단한 V/F 제어에는 필요하지 않은 솔루션 비용을 증가시킨다.

또한 PMSM 모터 제어 애플리케이션에 대한 구성 요소 사양을 정의하고 검증하기 위한 개발 시간도 필요하다. 개발자는 이러한 도전과제를 해결하기 위해 PMSM 모터 제어에 맞는 디바이스 사양과 고도의 아날로그 통합을 제공하는 마이크로컨트롤러를 선택할 수 있다. 이로써 필요한 외부 부품의 수를 줄이고 BOM(Bill of Material)을 최적화할

수 있다. 이제 고해상도를 가진 PWM과 함께 고도로 통합된 모터 제어 디바이스를 사용하여 고급 제어 알고리즘, 정밀 측정 및 신호 제어를 위한 고속 아날로그 주변장치, 기능 안전에 필요한 하드웨어 주변장치, 통신 및 디버그를 위한 직렬 인터페이스를 보다 손쉽게 구현할 수 있는 것이다.

또 하나 어려운 문제는 모터 제어 소프트웨어와 모터의 전자기계식 동작 간의 상호 작용이다. **그림 2**는 표준 센서리스 FOC 블록 다이어그램을 보여 준다. 이 개념을 실제 설계로 구현하기 위해서는 수학 집약적인(math-intensive) 시간임계형(time-critical) 제어 루프 구현을 위한 컨트롤러 아키텍처와 디지털 신호 프로세서(DSP) 명령에 대한 이해가 필요하다.

안정적인 성능을 달성하기 위해서는 제어 루프를 반드시 하나의 PWM 주기 내에서 실행해야 한다. 제어 루프의 시간 최적화가 필요한 이유는 세 가지로 정리할 수 있다.

- ① 제약 사항: 인버터 스위칭에서 발생하는 음향 노이즈를 줄이기 위해 20KHz(50 μ s 주기)나 그 이상에서 PWM스위칭 주파수를 사용해야 한다.
- ② 더 높은 대역폭의 제어 시스템을 구현하기 위해 반드시 하나의 PWM 주기 내에서 제어 루프를 실행해야 한다.
- ③ 시스템 모니터링, 애플리케이션별 기능 및 통신 등 다른 백그라운드 작업을 지원할 수 있도록 제어루프의 가동 속도가 빨라야 한다. 따라서 FOC 알고리즘을 10 μ s 미만에서 실행하는 것으로 목표를 잡아야 한다.

많은 제조사가 센서리스 회전자 위치 추정기(estimator)가 포함된 예제 FOC 소프트웨어를 제공하고 있다. 그러나 모터 회전을 시작하기 전 반드시 FOC알고리즘에 모터 및 하드웨어에 맞는 다양한 매개변수를 설정해야 한다. 필요한 속도 및 효율성 목표를 달성하기 위해서는 제어 매개변수 및 계수에 대한 추가 최적화 작업이 필요하다. 이 최적화 작업은 1) 모터

그림 1. 3상 전압원 인버터를 사용하는 3상 센서리스 PMSM 제어 시스템

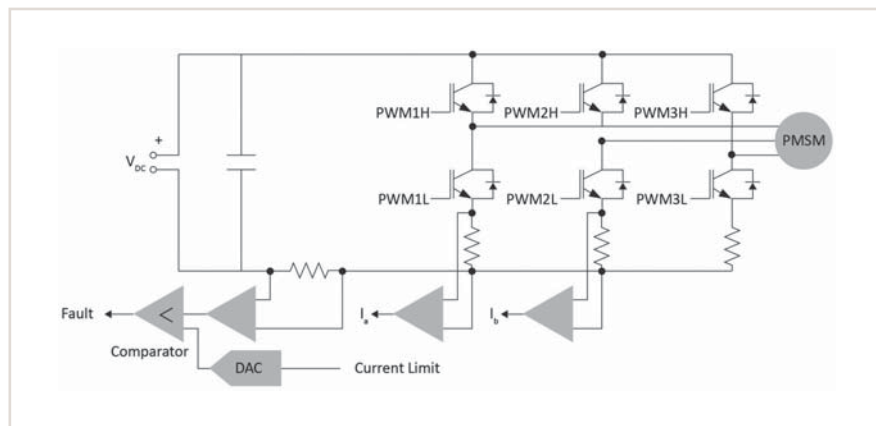
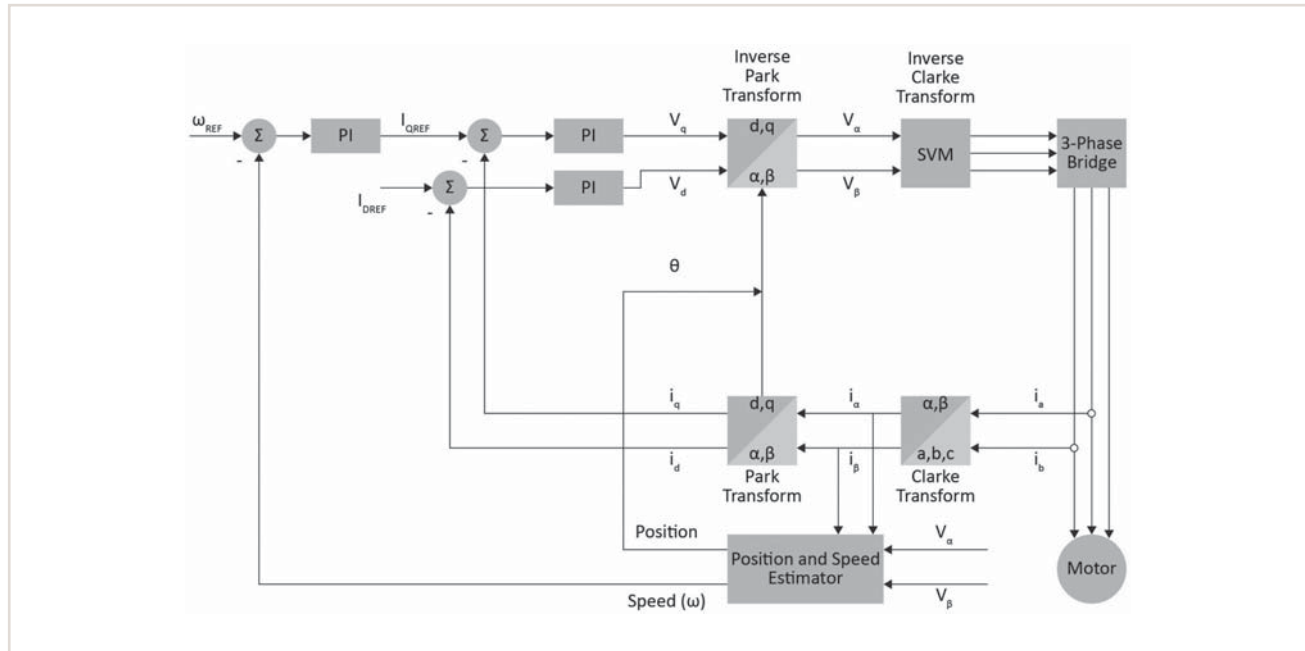


그림 2. 표준 센서리스 FOC 블록 다이어그램

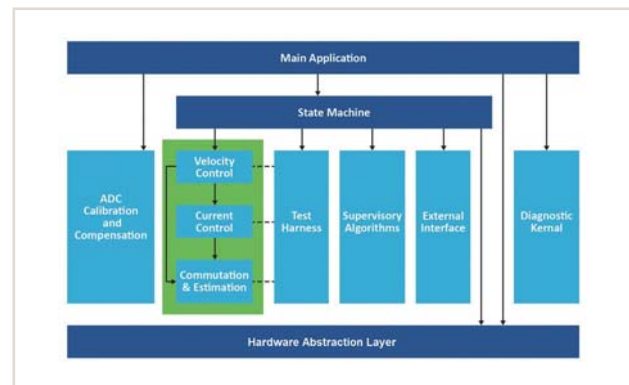


데이터시트를 통한 매개변수 유추와 2) 시행착오 방식을 통한 실험을 결합해 수행할 수 있다. 모터 데이터시트에 모터 매개변수가 정확하게 특정되지 않았거나 개발자가 고정밀 측정기에 접근할 수 없을 경우 개발자는 시행착오 방식에 의존할 수밖에 없다. 이러한 수작업 튜닝 과정에는 많은 시간과 경험이 필요하다.

PMSM 모터는 여러 애플리케이션에서 다양한 환경 및 설계 제약 하에 동작한다. 자동차 라디에이터 팬을 예로 들면, 모터가 시동되려는 시점에 팬 날개가 바람 때문에 역방향으로 자유롭게 회전할 때가 있다. 이런 상태에서는 센서리스 알고리즘이 적용된 PMSM 모터를 가동시키기란 어렵고, 인버터가 손상될 가능성도 있다. 한 가지 해결책은 회전 방향 및 회전자 위치를 감지한 뒤, 이 정보를 바탕으로 모터를 가동하기 전 능동 제동을 통해 모터를 천천히 정지시키는 것이다. 마찬가지로 단위 전류당 최대 토크(MTPA), 토크 보상, FW(Field Weakening)[1] 과 같은 추가 알고리즘을 실행해야 할 수도 있다. 실질적인 솔루션을 개발하기 위해서는 이러한 유형의 애플리케이션별 추가 알고리즘이 필요하지만, 이 경우 개발 시간이 늘어나고 소프트웨어 검증이 복잡해지는 등 설계 복잡성 역시 커진다.

이러한 복잡성을 줄이는 방법 중 하나는 시간이 중요한(time-critical) 작업에 영향을 미치지 않으면서 FOC 알고리즘에 애플리케이션별 알고리즘을 추가할 수 있는 모듈형 소프트웨어 아키텍처를 구축하는 것이다. 그림 3은 실시간 모터 제어 애플리케이션의 전형적인 소프트웨어 아키텍처를 보여주고 있다. 프레임워크의 중심에는 FOC 기능이 있고, 여기에는 엄격한 타이밍 제약과 다수의 애플리케이션별 추가 기능이 있다. 프레임워크 내 상태 머신(state machine)은 이러한 제어 기능을 메인 애플리케이션

그림 3. FOC 애플리케이션 프레임워크



과 연결한다. 아키텍처를 모듈형으로 만들고 코드 유지보수를 용이하게 하기 위해서는 잘 정의된 소프트웨어 기능 블록 간 인터페이스가 필요하다. 모듈형 프레임워크는 다른 시스템 모니터링, 보호 및 기능 안정 루틴과 함께 여러 가지 애플리케이션별 알고리즘의 통합을 지원한다.

모듈형 아키텍처의 또 다른 이점은 모터 제어 소프트웨어에서 주변 인터페이스 계층(또는 하드웨어 추상화 계층)이 분리되어 있으므로 애플리케이션 기능 및 성능 요건이 변경될 경우 개발자가 자신의 지적재산(IP)을 모터 컨트롤러 간에 원활하게 마이그레이션할 수 있다는 점이다.

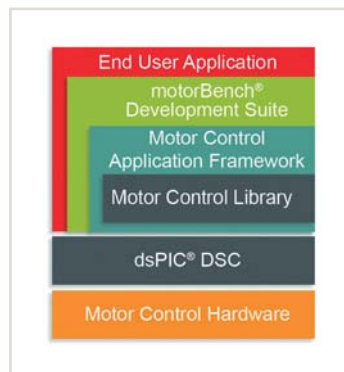
완전한 에코시스템 조성을 위한 요건

이러한 과제를 해결하기 위해서는 센서리스 FOC 설계에 맞춘 모터 제어 에코시스템이 필요하다. 고급 모터 제어 알고리즘 실행 프로세스를 간소화하기 위해서는 모터 컨트롤러, 하드웨어, 소프트웨어 및 개발 환경이 함께 동작해야 한다. 이를 위해서는 에코시스템에 다음과 같은 기능이 필요하다:

- ① 전문 지식이 없는 개발자도 모터 매개변수 측정을 자동화하고, 제어 루프를 설계하고, 소스 코드를 생성하는 고급 도구를 통해 FOC 모터 제어를 구현하고 시간이 많이 소요되는 복잡한 시간 결정적 코드를 작성 및 디버그할 수 있다.
- ② FOC용 애플리케이션 프레임워크 및 다양한 애플리케이션별 추가 알고리즘으로 개발 시간 및 테스트 시간을 단축할 수 있다.
- ③ 단일 칩에 신호 컨디셔닝 및 시스템 보호를 위한 확정적 응답 및 통합 아날로그 주변장치가 내장된 모터 컨트롤러를 사용해 총 솔루션 비용을 절감할 수 있다.

그림 4는 애플리케이션 프레임워크와 고성능 dsPIC33 모터 제어 DSC(디지털 신호 컨트롤러)를 위한 개발 제품군을 포함한 모터 제어 에코시스템 아키텍처의 예를 보여준다. 개발 제품군은 중요 모터 매개변수를 측정하고 피드백 제어 계인을 자동으로 조정할 수 있는 GUI 기반 FOC 소프트웨어 개발 툴을 기반으로 구축됐다. 또한 모터 제어 애플리케이션 프레임워크(MCAF)를 활용해 개발 환경에서 만

그림 4. 마이크로칩테크놀로지의 모터 제어 에코시스템 아키텍처



든 프로젝트의 소스 코드를 생성한다. 이 솔루션 스택의 중심에는 모터 제어 라이브러리(Motor Control Library)가 있다. 해당 라이브러리는 애플리케이션의 시간 임계형 제어 루프 기능을 실행하고 dsPIC33

DSC의 모터 제어 주변장치와의 상호작용을 지원한다. 해당 GUI는 여러 모터 제어 개발 보드와 함께 동작해 광범위한 저전압(LV) 및 고전압(HV) 모터에 대한 모터 매개변수 추출 및 FOC 코드 생성을 지원한다.

브러시리스 모터로의 전환 계기는 높은 에너지 효율성과 제품 차별화의 필요성에 있다. 포괄적인 모터 제어 에코시스템은 PMSM을 사용하여 센서리스 FOC 구현을 간소화하는 전체적인 접근방식을 제공하며, 코드 생성 자동화를 위한 전용 모터 컨트롤러, 신속한 프로토타이핑 개발 보드 및 사용하기 쉬운 FOC 개발 소프트웨어로 구성되어야 한다. **SM**

참고자료

- [1] 각도 추정 위상 고정 루프(PLL) 추정기를 사용하는 가전용 PMSM(표면 및 내부)의 FOC, TB3220
<http://www1.microchip.com/downloads/en/DeviceDoc/TB3220-Sensorless-Field-Oriented-Control-of-PMSM-for-Appliances-DS90003220A.pdf>
- [2] motorBench® 개발 스위트
<https://www.microchip.com/design-centers/motor-control-and-drive/motorbench-development-auto-tuning>
- [3] 모터 제어 디자인 리소스
<https://www.microchip.com/design-centers/motor-control-and-drive>
- [4] 모터 제어 라이브러리
<https://www.microchip.com/design-centers/motor-control-and-drive/motor-control-library>