

특집 : 최근의 전력전자 시뮬레이션 소프트웨어

PLECS로 전력전자 시스템에서의 열 및 전기적 특성 시뮬레이션

Wolfgang Hammer
(Plexim GmbH)

옮긴이 : 남 봉희(미림씨스콘 영업이사)

1. 서론

PLECS는 Simulink 환경에서 전기회로를 빠른 속도로 시뮬레이션 하기 위한 Tool box이고 전력전자와 드라이버 시스템에 중점을 둔 모든 전기회로 및 제어를 연동해서 시뮬레이션 할 수 있는 다목적 프로그램이다. 오늘날 전력전자 시스템은. 자동차, 항공, 의료, 가전, 에너지 관리, 신재생에너지 산업 등의 발전으로 새로운 회로와 개념을 빠르고 효율적으로 개발하고 평가할 수 있는 필요성이 요구되고 있다. 전력밀도와 효율을 요구하는 것이 증가함에 따라 전력전자 시스템에서 중요한 설계관점은 반도체의 전력 손실과 접합 온도를 적절히 추정하는 것이다. 사용자가 산업계나 학회에서 개발자, 연구원 또는 학생이든지 간에, PLECS는 전기시스템 디자인과 분석에 속도를 더해줄 것으로 기대하며 PLECS의 차별화된 특징들을 보면 아래와 같다.

2. Simulink와의 결합

2.1 회로 블록

PLECS에서 모델화된 각각의 전기회로는 Simulink내에서 개별 블록으로 표시되는데 전기적 소스 및 스위치소자를

제어하기 위해 신호를 블록으로 전달한다. PLECS 회로에서 수행된 측정치는 블록 출구를 통해 Simulink scope에서 볼 수 있으며, MATLAB에서 시뮬레이션이 구현되고, 시스템 제어에 사용할 수 있다. 또 모든 Simulink library를 사용할 수 있으며, 복잡한 제어기까지도 함께 모델링 할 수 있는 장점이 있다.

2.2 Simulink solver

시뮬레이션을 시작하면, PLECS회로는 플라이상에서 등가 방정식으로 변환되며, 이 방정식은 Simulink 모델을 따라 계산된다. Simulink와의 결합은 Simulink의 강력한 시뮬레이션 엔진을 모두 사용할 수 있는 혜택을 주고, Simulink의 zero-crossing detector 도 사용되는데, 이것은 스위칭 순간이 기계의 정밀도와 부합 된다는 확신을 주기 위한 것이다.

2.3 강력한 schematic editor

Simulink를 알면 PLECS schematic editor는 당장 사용할 수 있다. 서킷블록을 두 번 클릭하면 editor 창이 열리고, Simulink와 같이 쉽게 소자들을 끌어와 두고 연결하면 된다. 그리고 출판에 바로 사용해도 되는 선명한 schematic을 제공한다.

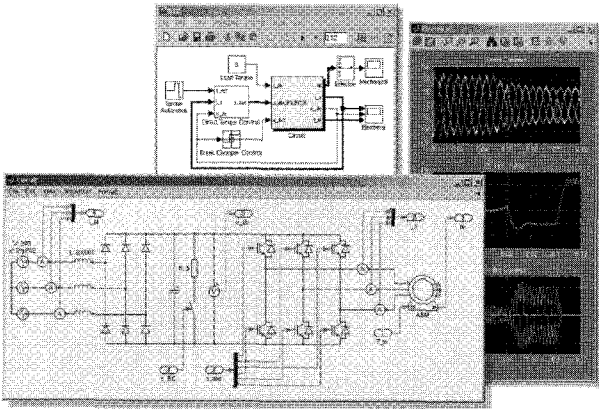


그림 1 전동기 구동장치의 경우 PLECS의 모델 및 시뮬레이션 예

2.4 직관적 모델링

회로의 인터페이스를 위해, 단지 입력단과 출력단을 끌어와 schematic에 붙이면 된다.

사실, PLECS block 동작은 아주 많은 부분, Simulink의 subsystem 같이 동작하고, 추가적으로 복잡함을 숨기거나 계속 사용하는 소자를 형성하기 위해 subcircuit을 사용할 수도 있다.

2.5 이상적 스위치 이용

PLECS에서, 전력전자 소자, 회로 차단기 등의 모델들은 이상적인 스위치로 해석한다. 이들은 실제 short-circuit($R_{on}=0$) 이거나 이상적 open circuit($R_{off}=\infty$)으로 표시되며, 스위칭은 순간적으로 일어난다. 모델링에 있어서 이상적 스위치를 사용하면 사용이 쉽고, 강건하며 속도가 빠르다는 장점을 제공한다.

2.6 사용의 편리성

이상적 스위치는 on 저항 값이나 스너버 캐패시터 값과 같이 고민해야 할 어떠한 parameter값이 필요없다. 특히 parasitic effects가 영향을 끼치지 않는 시스템 시뮬레이션에 있어서는 종종 이러한 값을 알지도 못할 것이다. 그러나, 아주 상세하게 특별한 전력전자 장비를 모델링 하고 싶다면, 필요에 따라 순방향 전압값이나 인덕턴스 등을 추가할 수 있다.

2.7 강인성

다른 시뮬레이션 프로그램에서 스너버 회로를 사용하면 시뮬레이션 모델이 아주 복잡해지고 어렵게 된다. 또 이러한 모델들은 fixed time step을 요구하거나 어려운 수치해석적 알고리즘을 필요로 한다. 반면 PLECS는 스너버없이 시뮬레이션 할 수 있으며, Simulink에서 제공되는 variable time step 또는 fixed time step 을 선택적으로 사용할 수 있다.

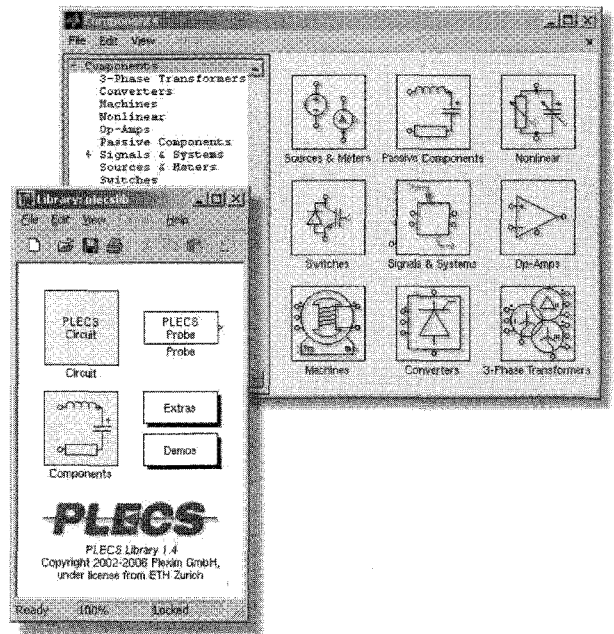


그림 2 PLECS의 Library 예

2.8 빠른 속도

기존의 회로 시뮬레이션 프로그램은 스위칭 순간을 계산하는데 아주 오랜 시간이 필요 하는데, 이것은 비선형 특성을 해석하기 위해 time step를 작게 쓰기 때문이다. 반면, PLECS에서는 이상적 스위치의 순간적 스위칭으로 이 문제가 해결되며, 한번의 스위칭 전후에 각 1번씩 총 2번의 step만을 필요로 한다.

2.9 Component library

Library를 보면, 전압소스, 전류 소스, 메터, 수동 소자들과 같은 표준 외에, PLECS는 다양한 반도체, 스위치, 회로차단기, 완성된 컨버터, 삼상 트랜스포머 등을 제공한다. 전동기 드라이버를 모델링하기 위해서, 하나의 AC 또는 DC machine를, 예를 들면 유도기거나 영구자석 동기기기를 가져오면 된다.

2.10 시스템 요구 사양

Windows, Mac OSX, Linux, and Solaris 에 사용가능하며, PLECS를 사용하기 위해 MATLAB 과 Simulink가 필요하다. (쥘리엥씨스콘에 요청하면 trial license를 무료로 받을 수 있다.

3. 전기적 특성과 열특성을 처리하는 방법

3.1 반도체 손실 추정

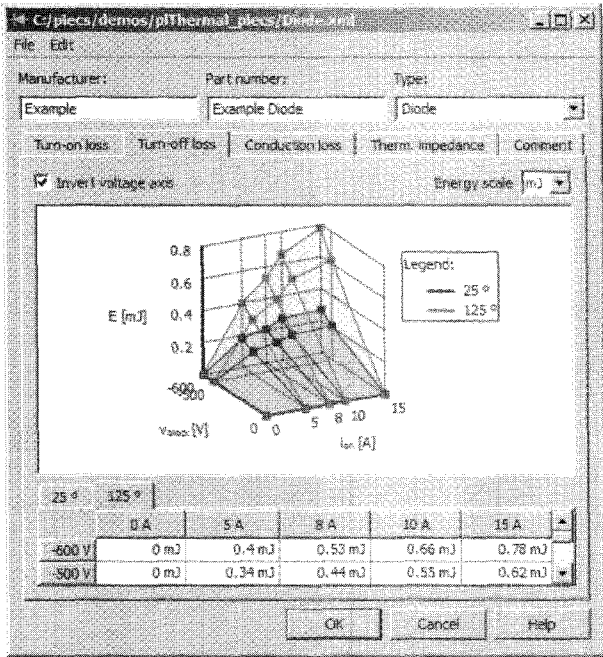


그림 3 열특성 에디터의 화면

전력반도체에 소비되는 손실을 계산하는 전통적 방식은 올바른 방법으로 보이기도 하지만 반도체를 가능한 한 상세하게 모델링하고 그 이후 소자의 전류와 전압강하를 단순히 곱하는 것이다. 그렇지만 이런 접근방법은 아래와 같은 심각한 결점을 가지고 있다.

1) 정확한 결과를 얻기 위해 Simulation은 Physical device model을 사용해야 하나, 전용 Software를 이용해야 할 뿐 아니라, 제조사에서 공개를 꺼리는 파라미터 들에 대한 지식을 필요로 한다(반도체 구조, Doping profiles등). 반면에 Behavioral device model은 손쉽게 사용할 수 있으나, 반도체 손실을 예상하기 위해 사용할 때는 신뢰성이 없다.

2) Switching transients를 상세하게 Simulation 하기 위해서는 마이크로 또는 심지어 나노 수준의 시뮬레이션 time-step이 필요하고, 반면에 전력전자 시스템의 열적 특성에서는 몇 분이나 되는 큰 시간상수 값을 가진다. 이것으로 결국 엄두도 못 낼 만큼 긴 Simulation 시간이 소요된다.

3.2 이상적 스위치와 look-up table 이용

이러한 이유로 PLECS는 다른 접근방법을 사용하고 있다. 전기적 영역에서 반도체 소자는 이상적으로 on/off를 수행하는 스위치로 단순화 하였고, 이것이 스위칭시 과도현상을 줄여 단 하나의 순간적 이벤트로 만든다. 스위칭 과도 상태 전후로, PLECS는 소자의 동작상태(forward current,

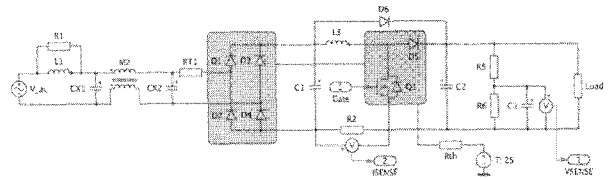


그림 4 SMPS의 회로 도면의 예

blocking voltage, device temperature)를 기록하고 이 파라미터를 사용하여 3차원 look-up table에서 불러온 소자의 소비된 에너지 결과를 읽는다. On-state 동안 소비된 전력은 소자의 전류와 온도로 계산된다. 필요한 data table들은 그림 3과 같이 PLECS에 내장된 Visual 편집기를 통해 입력할 수 있다.

이상화된 파형과 소자의 상세한 데이터를 결합하여 Simulation하는 이 방법은 아주 잘 알려진 것이며, 신뢰성 있는 결과를 만들기 위해 보인 것이다^[1]. 지금까지는 대개, 전기적 시스템의 시뮬레이션 후에 사용자가 직접 작성하는 Excel이나 Matlab등을 사용하여 후처리 단계에서 이행되었으며 이 접근법은 시간을 많이 소비하며, 에러가 발생하기 쉽고, 지속적으로 하기가 어렵다.

3.3 열특성 모델링

계산된 손실 값은 열 저항과 열 캐패시터 값으로 구성되는 등가회로인 열적 모델에 입력되고, 이때 반도체 소자는 열원으로 모델링 된다. 선택적으로 칩의 열 임피던스는 열적 RC chain으로 에디터에 입력될 수 있다. 그림 4는 부스트 토폴로지의 역률보상회로 PFC를 가지는 SMPS 회로를 보여주고 있으며 제어기는 Simulink에서 구현되며 여기에서는 나타나지 않았다. 청색 box는 이상적인 방열기로 표현되며, 태두리 내부 모든 반도체 소자로부터 소비되는 모든 손실을 흡수하고, 이들 소자들에 같은 온도 환경을 제공한다.

방열기는 자체의 thermal capacitance를 가지며 다른 thermal element와 연결될 수 있다. 그림 4에 있는 2개의 방열기는 하나의 방열판에 입력단 정류기와 PFC 단이 같이 장착되므로 직접 연결된다. thermal resistance를 통해 발열체와 외부의 온도가 연결된다. 그림 5는 일정한 부하조건에서 power supply를 구동시키는 동안의 과도시 파형을 보여주고 있다. DC 캐패시터에 최초 충전 이후 부스트 컨버터는 천천히 출력전압을 정격전압 값으로 끌어올린다. 약 150msec 시점에서 제어기는 dynamic mode로 변화하여 이것이 직접적으로 PFC 스위치의 온도에 반영된다. 그림에서 zoom 사진은 2번의 스위칭 주기동안 MOSFET의 온도를 나타낸 것인데, 각각의 스위칭마다 스위칭 손실의 순간적 소비에 따른 온도 상승을 보여준다. On-state 동안에는 온도는 지속적으로 계

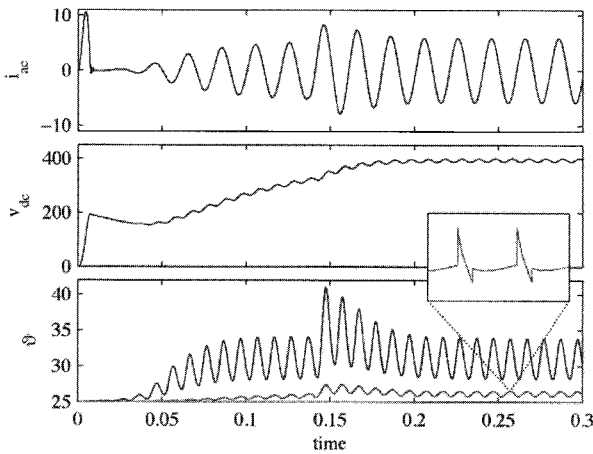


그림 5 전력 상승 과정의 과도 상태 파형
(상단: 주전류, 중간: 부하 전압, 하단: 발열체와 PFC 소자의 온도)

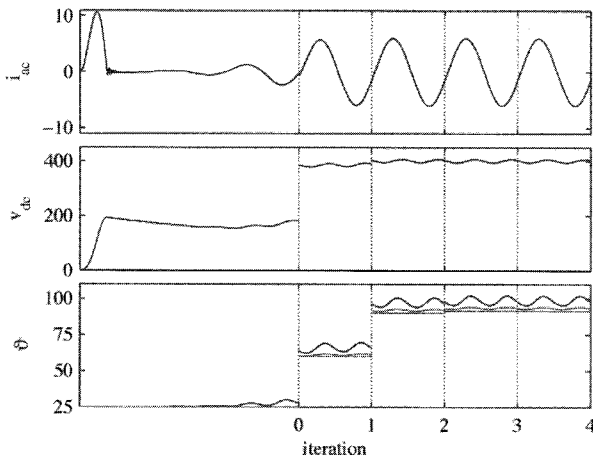


그림 6 정상상태의 도달 과정

속 올라가고, MOSFET이 차단되면 다시 온도는 내려간다.

3.4 정상 상태 해석

비록 그림 5에서는 보이지 않지만, 온도는 지속적으로 상승 중이고 상승세가 멈추는 최종의 온도를 알기 위해서는 비교적 빠른 PC를 사용하고, 이상적 스위치 모델을 사용한다 하더라도 시뮬레이션에 필요한 시간은 약 12시간이 될 것이다. 그러므로 PLECS는 스위칭 컨버터의 주기적인 정상상태 동작시점을 알아내기 위해 Iterative shooting 방법이라는 특징을 사용한다. 이런 방식의 정상상태 해석은 그림 6에서 보여

준다. 초기 몇 번의 사이클 이후 PLECS의 알고리즘은 초기 상태와 최종상태가 동일시되는 정상상태 조건에 더욱 잘 맞출 수 있도록 초기상태 값을 미리 알아낸다. 그림에서 보는 바와 같이, 3번의 iteration 이후에는 상대오차가 이미 1% 이하이며, 몇 번의 iteration이 더 지나면 상대오차는 약 10-5로 시뮬레이션이 끝난다. 이 모든 과정을 끝내는데 약 1분이면 된다.

4. 결론

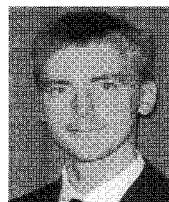
PLECS는 전력전자 시스템에서 열적 특성 시뮬레이션과 전기적 특성 시뮬레이션을 같이 할 수 있는 simulation tool이다. 소비된 소자의 손실 값을 알기 위해 이상 스위치 모델과 상세한 look-up table을 결합하여 사용하는데, 상세한 반도체 소자 시뮬레이션을 위한 효과적이고 정확한 방법을 제공한다. 규모가 큰 전력전자 시스템은 상세한 시뮬레이션을 위해 작은 시뮬레이션 time step을 사용 하므로 엄청나게 긴 시간을 필요로 하는데, PLECS 만이 실질적 도움이 되는 유일한 기술이다.

PLECS는 사용자가 직접 작성하고, 작업을 계속 유지하기가 힘든 열적 특성계산 tool을 대체할 수 있는, 사용자 위주의 통합된 환경을 제공한다.

참고 문헌

- [1] S. Munk-Nielsen, L.N. Tutelea, U.Jaeger, "Simulation with Ideal Switch Models Combined with Measured Loss Data Provides a Good Estimate of Power losses", Proc. IEEE Ind. App. Conference, Vol. 5, pp.2915-2922, 2000, Oct.

< 필 자 소 개 >



Dr. Wolfgang Hammer

1972년 12월 13일 독일 레버쿠젠 출생. 1997년 독일 아헨공과대학(RWTH Aachen) (석사). 1998년~2004년 스위스 ETH공과대학(ETH Zurich) 연구 및 조교 활동. 2003년 ETH공과대학(ETH Zurich) (공학). 현재 Plexim의 수석연구원 및 PLECS의 개발자로 활동.