

특집 : 전력전자 시뮬레이션 툴(1) PSIM

자동화 C 코드 생성방법

이 주 환

(한국파워심(주) 대표이사)

1. C 코드생성을 위한 SimCoder Module

1.1 개요

SimCoder는 PSIM에서 설계된 제어회로를 C코드로 자동으로 생성시켜주는 모듈이다. SimCoder를 사용하면, 시뮬레이션 작업과 연동하여 하드웨어를 손쉽게 구현할 수 있다. 이러한 CAE 기술을 이용함으로써 제품개발의 비용과 시간을 획기적으로 절감할 수 있다.

SimCoder는 범용의 하드웨어 타겟이 있는 경우와 없는 경우의 모듈을 지원한다. C 코드가 산출된 후에, 사용자는 일반적 형태의 하드웨어 플랫폼을 위한 코드로 바꿀 수 있는데, 메뉴에서 Elements → SimCoder for Code Generation → General Hardware 안에 일반 하드웨어를 위해서 제공되는 부품이 들어있다.

1.2 자동화 코드생성을 위한 부품

다음의 메뉴아래 포함된 모든 부품은 코드 생성을 위한 것이다. Elements → Event Control and Elements → SimCoder for Code Generation. 반면에 일반 PSIM library에 포함되어있는 몇몇 부품들 또한 자동화 코드생성을 위해 사용된다.

코드제너레이션에 사용되는 라이브러리와 일반 PSIM library와의 구별을 위하여 사용자가 구별할 수 있도록 표시해주고 있다. 메뉴에서 Options → Settings → Advanced에서 "Show image next to elements that can be used for code generation"의 옵션박스가 체크되어 있으면, PSIM 라이브러리의 오른쪽에 코드 생성을 위해 사용될 수

있다는 CG가 보여진다. 여기서 GH는 General HW Target 용을 의미한다.

2 자동화 코드생성 방법의 기초

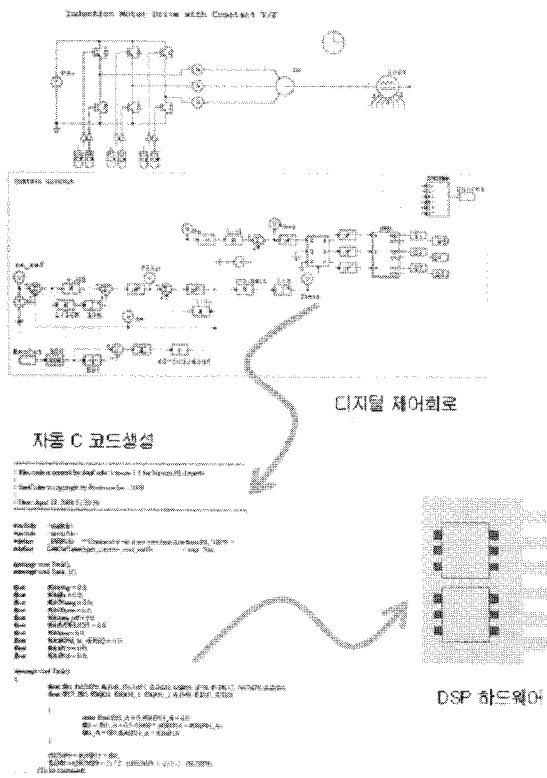
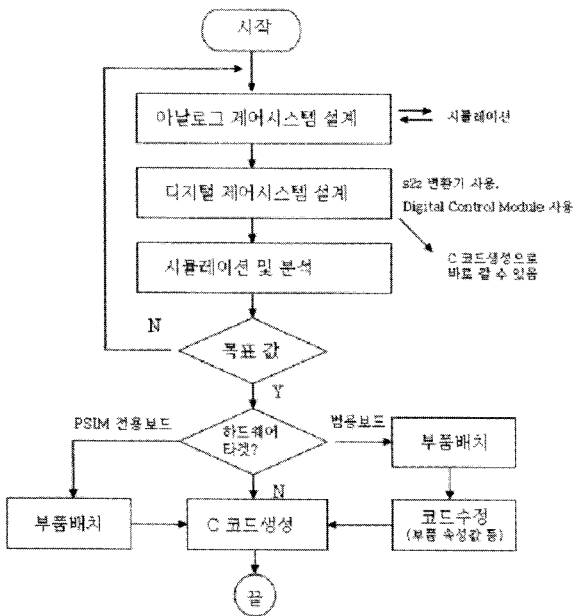
2.1 개요

일반적으로 SimCoder를 이용하여 코드생성을 하기위한 기본적인 단계는 다음과 같다.

- 아날로그 제어회로 시스템(continuous domain) 시뮬레이션
- 디지털 제어회로 시스템(discrete domain) 시뮬레이션
- 하드웨어 타겟이 없다면 서브서킷안에 제어회로를 배치한 후 코드를 생성한다.
- 하드웨어 타겟이 있으면, 하드웨어 요소를 포함시키는 시스템으로 수정한후에 코드를 생성한다. 물론, 시뮬레이션을 구동한후 그 결과를 확인할 수도 있다.

그러나 처음 두 번의 단계는 필수적인 것이 아니다. 예를 들어 PSIM으로 제어회로를 설계할 수 있다면 시스템을 시뮬레이션할 필요없이 바로 코드를 생성할 수 있다.

여기서 주의할 점은, 이산영역의(discrete domain) 디지털 제어회로에서만 코드를 생성할 수 있다. 즉, 아날로그(continuous domain)의 제어회로에서는 코드생성이 되지 않기 때문에, Digital control Module이 SimCoder에서는 필요하다. 간단한 예로 자동화 코드생성을 위한 절차를 도식화하면 다음과 같다.

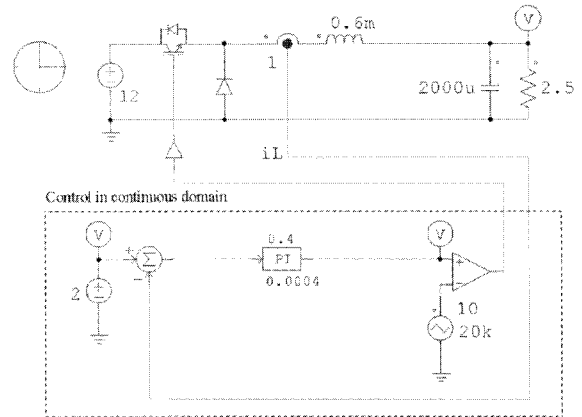


2.2 아날로그 제어회로 시스템

시스템을 설계할 때, 처음에는 아날로그 시스템으로 설계하고 시뮬레이션 해보는 것이 일반적인 순서이다. 여기서는 전류 feedback을 가지는 간단한 DC컨버터 회로를 예로 들어서 자세히 알아보기로 한다. 제어회로안에 있는 PI (proportional-integral) 제어기는 continuous s-domain으

로 설계되었다. 여기서, PI 게인 k 와 시정수 T 는 각각, $k = 0.4$, $T = 0.0004$ 이다. 스위칭 주파수는 20 kHz이다.

본 실습의 목적은, 점선안의 제어회로를 C 코드로 자동으로 생성하는 것이다. 코드생성을 실행하기 위한 첫 번째 단계는 s-domain안에 있는 아날로그 PI 제어기를 z-domain의 디지털 PI 제어기로 변환해야 한다.

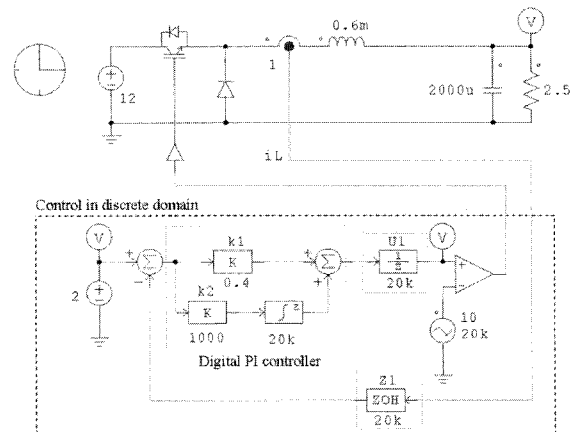


2.3 디지털 제어회로 시스템

아날로그 제어회로를 디지털 제어회로로 변화하기 위해서 Digital Control Module에서 제공하는 s2z Converter 유틸리티를 사용하면 쉽다. s2z 컨버터 유틸리티는 PSIM 메뉴에서 Utilities → s2z Converter를 선택하면 실행된다.

제일먼저, 아날로그 PI 제어기를 디지털 PI 제어기로 변환한다. 샘플링주파수는 스위칭주파수 20kHz와 같다. 한편, Digital Control 모듈에서 제공하는 s-to-z 변환 프로그램을 이용하면, 비례계수 $k_1=0.4$ 와 적분계수 $k_2=1000$ 를 가지는 디지털 PI 제어기를 얻을 수 있다.

완성된 디지털 제어회로는 다음의 그림과 같다. 여기서, 아날로그 제어회로와 비교하여, 회색박스안에 강조된 것과 같이, 세 개의 제어회로 블록으로 구성된다.



첫 번째로, 아날로그 PI 제어기는 디지털 PI 제어기로 대체되었다. 그리고 샘플링 주파수는 20kHz이며, 게인 k_1 과 k_2 는 s-to-z 프로그램으로부터 얻을 수 있다. 각각의 값을 그림에 표시하였다.

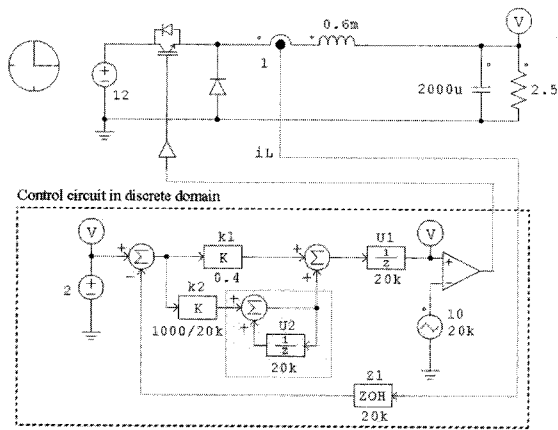
zero-order-hold 블록 Z_1 은 디지털 하드웨어 장비안의 A/D 컨버터에 의해서 샘플링된 피드백 전류 i_L 을 검출하기 위한 것이다.

unit-delay 블록 U_1 은 디지털 제어기에서 교유의 한 주기 지연을 표현하기 위한 것이다.

변환된 디지털제어기는 안정된 제어기 루프와 목표하는 결과치를 보여줘야 한다. 만일 디지털제어기의 시뮬레이션 결과가 안정되지 않거나 기대했던 결과가 아닐 경우에는, 다시 아날로그 제어회로 시스템으로 돌아가서 아날로그 제어기를 다시 설계해야 한다. 목표하는 결과를 얻기까지 상기의 과정을 반복하는 것이 매우 중요하다. 한편, 디지털제어회로에서의 적분기를 설계하기 위하여, 입출력 관계를 알아보면 다음과 같다.

$$y(n) = y(n-1) + T_s * u(n)$$

여기서, $y(n)$ 와 $u(n)$ 은 현재 시간의 출력과 입력이며, $y(n-1)$ 은 이전 샘플링주기의 출력이다. 그리고 T_s 는 샘플링주기이다. 위의 방정식을 이용하여 가산기와 unit-delay 블록을 가지고 적분기를 아래 그림처럼 설계할 수 있다.



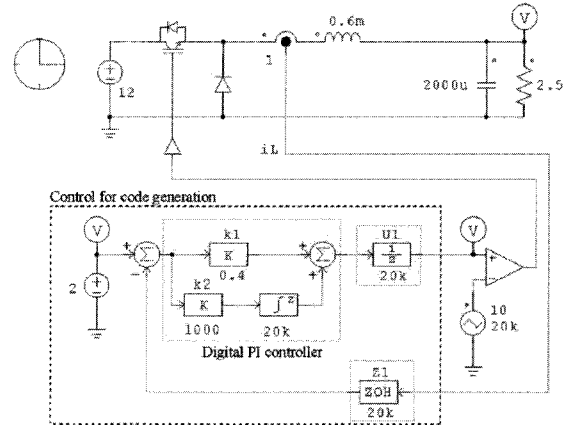
여기서, 비례게인도 샘플링시간 T_s 에 의하여 k_2 의 게인은 20kHz의 샘플링주파수가 추가된다.

이상으로 이산시스템의 디지털제어회로가 완성되었다. 이제 다음단계의 코드제너레이션을 수행할 준비를 해보자.

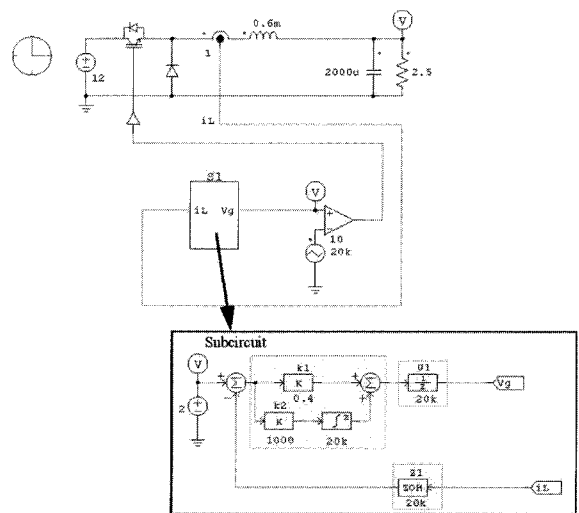
2.4 하드웨어 타겟이 없는 경우의 코드생성

SimCoder는 하드웨어 타겟 없이 시스템을 위한 코드를 생

성시킬 수 있다. 여기서, 설계자는 점선안에 있는 제어회로만을 위한 코드를 생성할 수 있다. 비교기와 삼각파발생기가 배제된 이유는 코드생성을 위해서는 스위칭주파수에 따른 샘플링레이트를 가진 부품으로 교체되어야 하는데, 이상은 하드웨어 타겟이 있는경우의 코드생성방법에서 자세히 다룬다.



점선안의 제어회로를 위한 코드를 생성하기 위해서는 제어회로를 서브서킷으로 만들어야 한다. 서브서킷을 만들기 위해서는, 마우스로 점선안의 회로를 선택한후, 마우스의 오른쪽 버튼을 클릭하면, 팝업메뉴가 나타난다. 팝업메뉴에서 Create Subcircuit을 선택하면 자동으로 서브서킷이 만들어지는데, 서브서킷의 파일이름을 입력하면 된다. 배선 및 입출력단자를 조정하여 배치한 최종 서브서킷은 아래그림과 같다.



서브서킷을 위한 코드를 생성하기위해서 다음의 과정을 따라한다. 주 회로에서 서브서킷을 선택하고, 마우스의 오른쪽 버튼을 클릭하면 팝업창이 나타나는데, 코드생성창을 띄우기

위한 Attributes을 선택한다. Subcircuit 창이 나타나면 Subcircuit Variables tab에서 Generate Code를 클릭하면, 자동으로 코드가 생성된다.

이와같이 자동으로 만들어진 코드를 발췌해보면 아래그림과 같다. 이 코드는 시뮬레이션을 위한 C block에서도 사용할 수 있다.

사용자는 코드의 앞단에 주석을 첨가할 수 있다. Simulation Control에서 SimCoder tab을 선택하면 다이얼로그 창이 나타나는데 여기에서 주석을 입력하고 편집하면 된다.

```

//*****
// This code is created by SimCoder Version 1.0
//
// SimCoder is copyright by Powersim, Inc., 2008
//
// Date: June 23, 2008 15:10:04
//*****

#include <stdio.h>
#include <math.h>
#define ANALOG_DIGIT_MID 0.5
#define INT_START_SAMPLING_RATE 1999999000L
#define NORM_START_SAMPLING_RATE 2000000000L

typedef void (*TimerIntFunc)(void);
typedef double DefaultType;
DefaultType *inAry = NULL, *outAry;
DefaultType *inFmErr = NULL, *outFmErr;

double fCurTime;
double GetCurTime() {return fCurTime;}

/* The C block for the generated code has the following additional output ports:
2 - S1:rcf
*/
void _SetVPS(int bRoutine, DefaultType fVal);
void fInfnOutArray()
{
    #if (1 > 0)
    inAry = new DefaultType[1];
    #else
    inAry = NULL;
    #endif
    #if (2 > 0)
    outAry = new DefaultType[2];
    #else
    outAry = NULL;
    #endif
}

void FreeInOutArray()
{
    if (inAry != NULL)
        delete[] inAry;
    if (outAry != NULL)
        delete[] outAry;
}
    
```

2.5 하드웨어 타겟이 있는 경우의 코드생성 방법

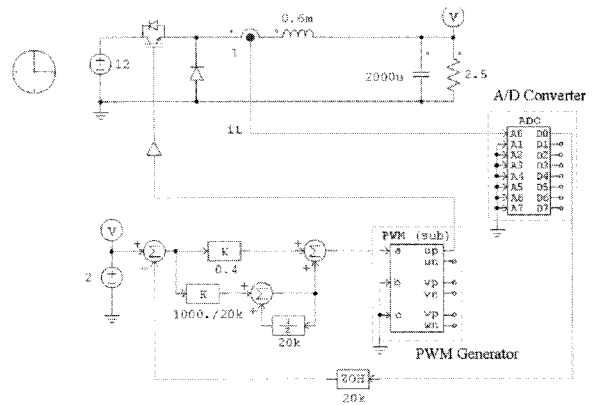
SimCoder는 특정한 하드웨어 타겟을 위한 코드를 자동으로 생성할 수 있다. 코드를 생성하기전에 제어회로 시스템에서 관련있는 하드웨어 부품을 포함시켜 수정해야한다. 또한 타겟 하드웨어의 입출력 값의 범위를 고려하여 정확하게 그 크기를 조정해야한다.

일반적으로, 다음의 부품 등이 추가된다.

- A/D Converters, Digital Input/Output 등의 추가.
- Hardware PWM generator를 포함한 PWM 발생기 재배치.
- 필요한 경우Event Sequence Control의 추가.

- Simulation Control 안에 hardware target의 지정.

사용자가 선택할 범용의 하드웨어 타겟을 사용하는 경우에는 PSIM 메뉴의 Elements → Simcoder for Code Generation → General Hardware 의 부품을 사용하여 배치하면 된다. 아래 그림은 타겟 하드웨어 부품을 배치한 후의 회로이다. 더 나은 도면의 이해를 위해 하드웨어 요소를 회색 박스안에 표시하였다.



디지털시스템 제어회로와 비교하여 다음과 같은 차이가 있다.

- 전류센서 출력에 A/D converter가 추가된다. 여기서, A/D converter의 입력범위의 설정에 주의해야 한다. 만일 전류센서의 출력이 A/D converter의 범위를 넘어서지 않도록 조절해야한다.
- 이전 회로에서 PWM 파형을 위한 비교기와 삼각파발생기는 하드웨어 PWM 발생기로 대체된다. 여기서, 하드웨어 PWM 발생기에는 한주기 지연된 샘플링을 포함하고 있기때문에, 원래 회로의 비교기 입력값에 있었던 unit delay 블럭 은 더 이상 필요치않다. PWM 발생기에서의 속성값 Carrier Frequency는 20 kHz에 맞춰지며, Peak Value 는 10V이다. 이는 이전회로의 삼각파발생기와 같다. 그리고 "Start PWM at Beginning"의 값을 Start를 선택한다.
- Simulation Control에서의 속성값 Hardware Type 은 사용하는 타겟 보드에 따라서 선택하면 된다. 범용의 DSP 타겟보드를 사용할 경우에는 General Hardware를 선택하면 된다.

한편, 사용자는 자동으로 생성된 코드의 초기에 주석란을 추가할 수 있다. Simulation Control에서 주석을 만들고 편집하기 위해 SimCoder tab으로 이동하여 다이얼로그 창에서

주석을 입력하고 편집하면 된다.

타겟 보드에서 하드웨어 부품을 추가하고 시스템의 동작을 검증하기 위해 시뮬레이션해 볼 수 있다. 시뮬레이션 결과는 앞에서 다룬 디지털제어회로의 시뮬레이션 결과와 매우 비슷할 것이다. 한번 시뮬레이션 결과가 검증되면, C 코드는 PSIM 메뉴에서 Simulate → Generate Code 를 선택하여 자동으로 코드를 생성할 수 있다. PSIM 라이브러리에 삽입된 전용의 보드가 있는 경우에는 수정 없이 바로 구동할 수 있고, 범용의 하드웨어 타겟을 선택한 경우에는 그 하드웨어의 요소에 맞는 입출력의 속성값 등을 수정하여 구동하면 된다.

이상에서 코드생성을 위한 방법을 순차적으로 자세히 설명하였다. PSIM의 Add-on 모듈 SimCoder(R)를 이용하면, 아주 쉽게 디지털제어회로의 C 코드를 만들 수 있으며, 여러분의 DSP 하드웨어 타겟에 바로 적용시킬 수 있다. 지금 바로 여러분의 프로젝트에 적용해 보시길 바란다. ■

〈 필 자 소 개 〉



이주환(李周煥)

2008년 현재 한국파워심(주) 대표이사.