

## 온도 제어기 설계를 위한 유한 요소법을 이용한 시뮬레이션 환경 프로토타입 구현

장유진\*

<sup>1</sup>동국대학교 정보통신학부

### Development of a Prototype of FEM Simulation Environment for Temperature Controller Design

Yu Jin Jang<sup>1\*</sup>

<sup>1</sup>Department of Information & Communication Engineering, Dongguk University

**요 약** 온도 제어를 필요로 하는 많은 산업 응용분야에서는 통상적으로 온도제어 대상물체를 특정 목표온도에 매우 가깝게 도달하도록 히터들의 출력을 조절한다. 제어 대상물체의 매우 느린 온도변화 반응과 이웃한 히터들에 의해 발생하는 열 간섭 현상은 다양한 산업 응용분야에서 요구되는 성능을 충족시키는 온도 제어기 설계를 매우 어렵게 하는 요인이며 많은 연구가 지속적으로 이루어지고 있다. 그러나 온도 제어기 설계를 위해서는 반복적인 실험이 매우 중요하지만 대상물체의 가열과 냉각에 매우 많은 시간이 소모되는 개선될 수 없는 현실이 존재한다. 따라서 대상물체의 온도 변화에 대한 수치해석 결과를 적절한 시간 내에 충분히 정확하게 나타낼 수 있으며 제어기 설계에 편리하게 응용 할 수 있는 시뮬레이션 환경이 필요하다. 본 논문에서는 온도 제어를 필요로 하는 산업분야에 유용하게 적용될 수 있는 유한요소법(FEM: Finite Element Method)을 이용한 2D 시뮬레이션 환경의 프로토타입을 MATLAB을 이용해서 구현했다. 개발된 프로토타입은 기본적인 물체 외형 디자인, 추가 경계 설정, 메쉬 생성 및 FEM 수치해석 기능을 갖추고 있으며 예제를 통하여 동작의 유용함과 편리함을 보였다.

**Abstract** In many industrial applications, it is very important to control the temperature of the controlled object to the target temperature as closely as possible. Although it is apparent that the great obstacles in controller design are time-delay of the thermal responses of the controlled object and the effect of thermal interference between neighboring heating zones, one more fundamental obstacle is a very large amount of time which is required during repeated experiments in controller design process. Therefore, a convenient simulation environment, which can represent thermal behavior accurately within appropriate time, is needed. In this paper, a prototype of 2D FEM (finite element method) heat transfer simulation environment using MATLAB is constructed to be usefully adopted into industrial applications with temperature controller design.

**Key Words** : Prototype, simulation environment, heat transfer, mesh generation, FEM, MATLAB

### 1. 서론

다양한 산업현장에서 여러 개의 히터(heater)를 사용하여 물체의 온도를 일정한 목표 온도로 조절해야 하는 경우가 흔히 존재하지만 보통 히터 출력 변화에 대한 매우 느린 온도변화와 이웃 영역간의 열 간섭(thermal interference) 현상으로 인해 빠른 온도 제어가 매우 어렵

다. 이러한 문제를 해결하기 위해 다양한 방법들이 연구되고 있으며 반복적인 실험을 통한 모델링 및 설계된 제어기 검증과정이 요구된다. 새로운 아이디어를 적용한 제어기 설계를 위해서는 보통 많은 실험이 필요하지만 실험의 특성상 가열과 냉각에 매우 많은 시간이 소모되는 큰 문제가 존재한다. 또한 새로운 제어기 설계를 위해서는 필요한 위치의 다양한 온도 데이터를 수집할 수 있어

\*교신저자 : 장유진 (season@dongguk.ac.kr)

접수일 09년 10월 09일

수정일 10년 02월 09일

게재확정일 10년 02월 24일

야 하지만 필요시마다 온도 센서를 추가로 설치하여 실험을 진행하는 것은 매우 많은 비용과 시간을 필요로 한다. 이러한 문제를 해결하기 위해 시뮬레이션을 도입하는 것은 합리적인 대안이다. 이러한 시뮬레이션은 열전달 현상을 사실적으로 묘사할 수 있어야 하며 보통 편미분 방정식을 포함하는 다양한 공학문제들을 효과적으로 분석하게 도와주는 수치해석 방법인 유한요소법(FEM: Finite Element Method)이 사용된다 [1-3]. 이러한 FEM은 컴퓨터 성능향상에 힘입어 목적하는 제품 또는 시스템의 성능 향상을 위해 제어공학 분야와 연계되어 적용되고 있는 추세이다 [4]. 예를 들어 [5]에서는 시간지연이 거의 없는 매우 얇은 평판에 대해 두 개의 히터를 이용하여 온도제어를 수행했으며 이 과정에서 히터들 간의 열 간섭 현상을 명시적으로 모델링했으며 독특한 디커플링(decoupling)기법의 효용성을 FEM 시뮬레이션을 통해 검증했다. 그러나 이는 열전달 응답의 데드타임(dead time)이 거의 없는 상황에 대한 논의이며 긴 데드타임을 갖는 상황에 대한 문제는 여전히 해결해야 할 문제이다.

문제의 특성에 따라 많은 특화된 FEM 소프트웨어들과 범용 FEM 소프트웨어가 존재하며 제어기 설계에 응용되기 위해서는 외부 프로그램과 상호 작용할 수 있는 통로가 있어야만 한다. 그러나 이러한 기능이 제공되지 않는 소프트웨어들도 존재하며 열전달 문제로 국한하면 비열(specific heat)과 열전도도(thermal conductivity)의 온도에 따른 비선형성을 허용하지 않는 소프트웨어들도 존재한다.

제어기 개발 단계에서는 편리하게 복잡한 수학적 알고리즘들을 디자인 할 수 있는 개발 환경이 매우 중요하며 MATLAB이 많이 사용되고 있다. FEMLAB [6], Structural Dynamic Toolbox [7], Visual NASTRAN 4D [8] 등은 MATLAB과의 연동기능을 제공하는 상용 FEM 소프트웨어이지만 Visual NASTRAN 4D에서는 정상상태(steady state) 열전달 해석만 가능하다. 상용 FEM 소프트웨어의 비용 및 기능, 사용자 목적에 따른 제어기 설계의 편리함 등 여러 사항을 고려했을 때 직접 FEM 코딩을 수행해야만 하는 경우가 종종 있다. 본 논문에서는 FEM을 이용하여 편리하게 온도제어기 설계에 응용할 수 있게 하기 위해 MATLAB에서 2D FEM 열전달 시뮬레이션 환경의 프로토타입(prototype)을 구현했다. 이를 위해서는 대상 물체의 외형을 디자인하는 기능과 디자인된 외형을 유한개의 요소로 나누는 메쉬(mesh) 생성 기능 및 생성된 메쉬를 이용한 수치해석 기능이 필요하다. 또한 사용자의 목적에 따른 손쉬운 코드 변형 및 적용을 고려하면 매우 간단한 코드로 이루어져야만 한다. 공개된 여러 소스들을 조합하고 변형하여 이러한 기능을 구현했으며 두 개의

히터를 이용하여 물체를 가열하는 예제를 통하여 제안된 시뮬레이션 환경의 유용성을 보였다.

## 2. 시뮬레이션 환경 구현 및 테스트

### 2.1 물체 외형 디자인 및 추가 경계 생성

대부분의 상용 FEM 소프트웨어에서 물체 외형 디자인을 위하여 GUI(graphic user interface) 환경을 기본으로 제공하고 있으며 FEMLAB과 ANSYS에서와 같이 간단한 명령을 입력하여 외형을 디자인하는 CUI(command user interface) 환경도 같이 제공하는 소프트웨어들도 있다. 이러한 CUI 기반의 모델링 기능은 외형을 일정한 기준에 따라 변화시키면서 시뮬레이션을 수행해야 하는 상황에서 매우 유용하게 사용될 수 있으며 [9]에서는 로봇 외형을 최적화 시키는 문제를 ANSYS의 APDL[10] 이라는 스크립트 언어의 기능을 이용하여 해결하였다.

본 논문에서는 이러한 점을 고려하여 CUI 기반의 외형 디자인을 채택하였다. 매우 간단하게 물체의 외형을 표현하기 위해 외형은 점들을 잇는 직선들의 조합으로 이루어진 폴리곤으로 표현했다. MATLAB에서 선, 원, 사각형, 삼각형 등의 기본적인 외형 생성 및 회전 기능을 갖는 함수를 간단하게 작성할 수 있으며 이들 폴리곤 간의 연산(교집합, 합집합, 차집합, 배타적 논리합)을 이용한 외형 생성은 공개 소프트웨어인 PolygonClip 함수를 이용하여 간단히 구현했다 [11]. 디자인된 외형에 히터 위치 설정 등을 위한 경계 설정이 필요하며 직접 외형을 나타내는 자료 구조상에 추가 할 수도 있으며 GUI 환경에서 마우스를 이용하여 표시할 수도 있다. GUI 환경에서의 기능은 화면 확대 및 축소기능이 편리한 공개 소프트웨어인 [12]를 변형하여 구현하였다. 간단한 외형 디자인 및 경계 추가의 예를 표 1과 같이 그림 1에 나타내었다. 표 1은 5개의 원을 생성하고 이들 간의 차집합 연산을 수행하여 그림 1과 같은 외형을 손쉽게 생성하는 코드를 나타내고 있다. 그림 1에서는 전술한 GUI 환경에서의 추가적인 경계 설정도 보여주고 있다. 그림 1에서 좌측의 2개의 빨간색 원은 사용자가 선택한 경계 위치를 나타내고 있으며 메쉬(mesh) 생성 시 이 위치에 노드(node)를 고정 시키고 해석 수행에 필요한 경계 조건을 설정할 것이다.

### 2.2 메쉬생성

유한요소 해석을 위해서는 디자인된 외형을 유한개의 요소로 나누는 메쉬 생성 과정이 반드시 필요하지만 공개된 MATLAB용 메쉬 생성 코드는 매우 드물다. [13]에

서는 힘의 평형 원리를 이용한 매우 간단한 MATLAB용 메쉬 생성 기법과 코드를 공개하고 있으며 레벨 셋(level set) 개념을 이용하여 물체를 표현하고 있다. 물체의 외형(경계)은 제로 레벨 셋(zero level set)으로 정의하고 외형과 특정 위치까지의 최단 거리와 ± 부호를 이용하여 내부는 0보다 작은 값으로, 외부는 0보다 큰 값으로 정의하는 implicit geometry 개념을 사용하고 있다. 따라서 2.1절에서 기술한 explicit geometry 개념을 이용한 외형 디자인 방법과는 맞지 않기 때문에 본 논문에서는 [13]에서 제안한 메쉬 생성 기법을 사용할 수 있도록 코드를 수정하였다.

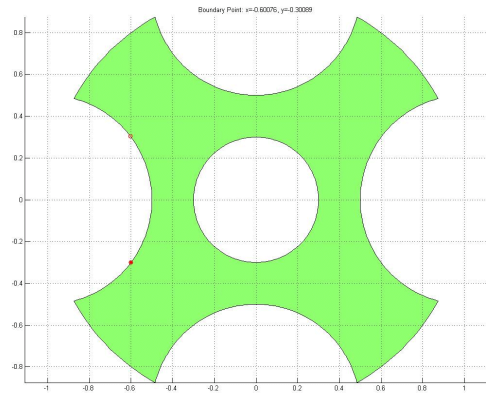
생성된 메쉬들을 이용하여 FEM에서 편미분 방정식의 해를 계산할 때 메쉬들이 정삼각형에 가까울수록 수치해석 결과가 좋다고 알려져 있으며 메쉬 품질은 [13]에서와 같이 정삼각형에서는 1의 값을 가지고 삼각형을 이루지 못했을 때 0의 값을 가지는 인덱스를 사용하였다. 모든 삼각형에 대해 최소 품질이 0.5 이상이면 적절하다고 알려져 있다. 그림 1의 외형과 추가 경계점 정보를 바탕으로 메쉬 생성을 수행한 결과를 그림 2에 나타냈으며 메쉬 최소 품질은 0.7396으로 적절한 수준이다. 균질한 메쉬(uniform mesh) 대신 외형의 여러 특징 등을 이용하여 삼각형의 크기를 다양하게 변화시킬 수도 있지만 사용하지는 않았다. [13]은 일반적으로 높은 품질의 메쉬를 생성해 내지만 많은 계산 시간을 필요로 하는 단점이 있다. [14]에서는 현재까지 알려진 메쉬 생성기법을 체계적으로 소개하고 있으며 필요시 이를 참고하여 적절한 메쉬 생성 알고리즘을 디자인하여 사용할 수도 있을 것이다.

### 2.3 열전달 해석

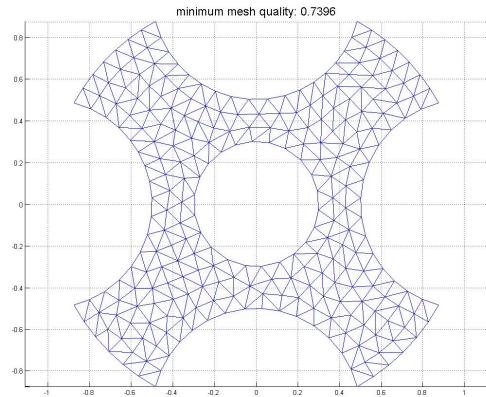
표 2와 같이 기호를 정의했을 때 2D 열전달 방정식은 식 (1)과 같이 편미분 방정식으로 표현된다 [1-2].

[표 1] 외형 디자인 예제 코드 (MATLAB)

make_circle(x, y, r)	중심(x, y), 반지름 r인 원
P(1) = make_circle(0, 0, 1);	
P(2) = make_circle(1, 0, 0.5);	
P(3) = make_circle(-1, 0, 0.5);	
P(4) = make_circle(0, 1, 0.5);	
P(5) = make_circle(0, -1, 0.5);	
P(6) = make_circle(0, 0, 0.3);	
PolygonClip(A, B, 0)	A-B 연산
S = PolygonClip(P(1), P(2), 0);	
S = PolygonClip(S, P(3), 0);	
S = PolygonClip(S, P(4), 0);	
S = PolygonClip(S, P(5), 0);	
S = PolygonClip(S, P(6), 0);	



[그림 1] 외형 디자인 결과 및 추가 경계 설정



[그림 2] 메쉬 생성 (최소 메쉬 품질: 0.7396)

$$\rho C(T) \frac{\partial T}{\partial t} = \frac{\partial}{\partial x} \left( k(T) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k(T) \frac{\partial T}{\partial y} \right) \quad (1)$$

식 (1)에서 비열과 열전도도는 일반적으로 온도에 대한 함수이다.  $\alpha = k/\rho C$ 로 정의하고 식 (1)의 양변을  $\rho C$ 로 나누고 2D 도메인  $\Omega$ 에 대해서 weighted residual form을 식 (2)와 같이 얻을 수 있다.

$$\int_{\Omega} W \left( \frac{\partial T}{\partial t} - \frac{\partial}{\partial x} \left( \alpha \frac{\partial T}{\partial x} \right) - \frac{\partial}{\partial y} \left( \alpha \frac{\partial T}{\partial y} \right) \right) d\Omega = 0 \quad (2)$$

식 (2)에 Green's Theorem을 적용하면 식 (3)과 같은 형태로 변형된다 [1-2].

$$\int_{\Omega} \left[ W \frac{\partial T}{\partial t} + \alpha \left( \frac{\partial W}{\partial x} \frac{\partial T}{\partial x} + \frac{\partial W}{\partial y} \frac{\partial T}{\partial y} \right) \right] d\Omega \quad (3)$$

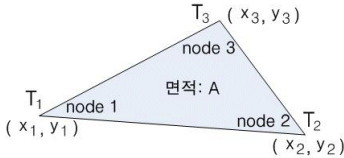
$$+ \int_{\Gamma_B} \frac{1}{\rho C} W \left( -k \frac{\partial T}{\partial n} \right) d\Gamma = 0$$

식 (3)에서  $\Gamma_B$ 는 디자인된 외형의 경계 중에서 플럭스(flux) 형태의 Neuman 경계 조건이 적용되는 부분이며  $n$ 은 경계에서 바깥 방향으로 수직(outward normal)인 단위 벡터를 나타낸다. 식 (3)의  $-k \frac{\partial T}{\partial n}$ 은 상수 플럭스 및 대류, 복사 열전달을 고려하여 다음 식(4)와 같이 표현된다 [1-3].

$$-k \frac{\partial T}{\partial n} = -q + h(T - T_{ambient}) + 5.67 \times 10^{-8} \epsilon (T^4 - T_{ambient}^4) \quad (4)$$

식(4)에서  $q$ 는 경계에서 내부로 가해지는 플럭스를 나타내며  $T_{ambient}$ 는 주변 공기의 온도를 나타낸다.  $h$ 와  $\epsilon$ 은 각각 대류 열전달 계수와 방사율(emissivity)을 나타낸다.

그림 3과 같이 linear triangular element를 사용하면 각 노드에서의 온도값을 이용하여 삼각형 내부의 임의의 위치의 온도를 다음 식 (5)와 같이 나타낼 수 있다 [1-3].



[그림 3] 선형 삼각형 요소

[표 2] 사용된 기호 및 의미

변수	의미 및 단위	변수	의미 및 단위
$x, y$	$x, y$ 좌표 [m]	$\rho$	밀도 [ $kg/m^3$ ]
$T$	온도 [K]	$C$	비열 [ $J/kg K$ ]
$t$	시간 [sec]	$k$	열전도도 [ $W/mK$ ]

$$T(x, y, t) = \sum_{i=1}^3 H_i(x, y) T_i(t)$$

$$H_1 = \frac{1}{2A} ((x_2 y_3 - x_3 y_2) + (y_2 - y_3)x + (x_3 - x_2)y)$$

$$H_2 = \frac{1}{2A} ((x_3 y_1 - x_1 y_3) + (y_3 - y_1)x + (x_1 - x_3)y)$$

$$H_3 = \frac{1}{2A} ((x_1 y_2 - x_2 y_1) + (y_1 - y_2)x + (x_2 - x_1)y)$$

(5)

식 (5)를 식 (3)에 대입하고 각각의 삼각형 요소에 대해  $W_i = H_i, (i=1,2,3)$ 로 설정하는 Galerkin method를 적용하면 최종적으로 식(6)과 같은 형태로 나타낼 수 있다 [1-3].

$$M \frac{dT}{dt} + KT = F \quad (6)$$

식 (4)에서 상수 플럭스  $q$ 와 대류 열전달을 나타내는  $h(T - T_{ambient})$ 만 고려할 때는 식 (6)과 같이 바로 나타낼 수 있지만 복사 열전달을 나타내는  $5.67 \times 10^{-8} \epsilon (T^4 - T_{ambient}^4)$  항이 포함되면 곤란하게 된다. 이때는 복사열전달 계수  $h_r$ 을 식 (7)과 같이 정의하면 대류항과 같은 형태가 되어 손쉽게 해결할 수 있으며 반복적으로 온도  $T$ 를 구하고 이를 이용하여 다시  $h_r$ 을 계산하여  $h_r$ 이 적절히 수렴할 때까지 계산을 수행한다 [15].

$$5.67 \times 10^{-8} \epsilon (T^4 - T_{ambient}^4)$$

$$= 5.67 \times 10^{-8} \epsilon (T^2 + T_{ambient}^2)(T + T_{ambient})(T - T_{ambient})$$

$$= h_r (T - T_{ambient}) \quad (7)$$

시뮬레이션의 시간 간격을  $\Delta t$ 라 하고  $F$ 와  $T$ 에 동일한 보간법(interpolation)을 사용하면 시간 진행에 따른 해는 식 (8)과 같이 나타낼 수 있다 [2].

$$\left( \frac{M}{\Delta t} + K\theta \right) T^{n+1} = \left( \frac{M}{\Delta t} - K(1-\theta) \right) T^n + \theta F^n + (1-\theta) F^{n+1} \quad (8)$$

식 (8)에서 첨자  $n$ 과  $n+1$ 은 각각  $n$ 과  $n+1$ 번째 샘플을 의미하며  $\theta=1/2$ 일 때는 Crank- Nicholson 기법을 나타내며,  $\theta=2/3$  일 때는 Galerkin 기법을 나타낸다. 두 기법 모두 항상 항상 안정적(unconditionally stable)이지만 Galerkin 기법이 에러의 진동이 작은 것으로 알려져 있다 [2]. 해를 구하기 위해서는 코딩작업을 수행해야 하며 본 논문에서는 [1-3]을 참고하여 [3]에서 제공하는 소스 코드를 변형하여 MATLAB용 FEM 코드를 작성하였다. 이를 위하여 생성된 요소들의 노드를 모두 반 시계방향으로 정렬하는 기능, 경계 구분 및 경계에 속한 노드들을 순서대로 골라내는 기능, 경계조건을 적용하는 기능, 특정 위치를 포함하는 요소를 골라내는 기능 등도 구현하였다.

### 2.4 시뮬레이션

본 절에서는 제안된 시뮬레이션 환경의 유용함을 두 개의 히터를 이용하여 물체를 가열하는 예를 통해 보였다. 지름 15cm, 길이 50cm의 원기둥 모양의 물체를 두 개의 밴드타입 히터로 둘레를 감싸고 가열하는 상황을 가정하자. 이 문제는 2D로 표현 가능하며 그림 4에서와 같이 밴드 폭이 10cm이며 와트 밀도가 4 W/cm<sup>2</sup>인 두 개의 히터가 5cm 간격을 두고 설치되어있다. 또한 히터가 설치되어 있지 않은 경계에서의 열전달은 대류 열전달만을 고려하자. 물체의 밀도, 비열, 열전도도, 대류 열전달 계수, 물체와 공기의 초기온도를 표 3에 나타내었다.

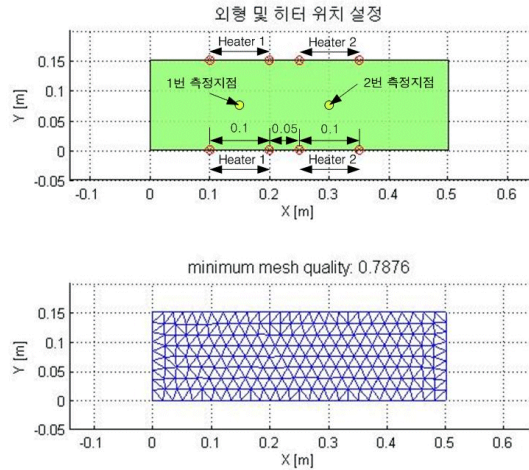
먼저 히터가 설치될 곳의 위치를 고려하여 제안된 소프트웨어를 사용하여 물체의 외형을 디자인 하고 총 404 개의 요소를 생성하였다. 또한 메쉬의 품질을 나타내는 지수가 0.7876으로 적절함을 알 수 있다.

히터는 대부분 0에서 1의 값을 가지는 듀티 비(duty ratio)를 이용하여 그림 5와 같이 한 주기 동안 켜지는 시간과 꺼지는 시간을 조절하는 기본적인 ON/OFF 제어를 통해 동작하며 제어기에 의해서 듀티 비를 조절하게 된다. 이러한 시스템에서는 보통 한 주기마다 온도 데이터를 수집하고 히터를 제어하게 된다. FEM을 이용해서 각 히터의 듀티 비에 따른 온도 변화를 시뮬레이션 하기 위해서는 한 주기 동안 각 히터들의 ON/OFF 상태와 시간 간격이 필요하다. 그림 6에서는 이러한 상황의 일부를 표현했으며 히터 1의 듀티 비가 히터 2의 듀티 비보다 크기 때문에 한 주기 동안  $\Delta t_1$ ,  $\Delta t_2$ ,  $\Delta t_3$ 의 시간 간격마다 각 히터들의 ON/OFF 상태를 고려하여 FEM을 적용해야만 한다. 히터가 두 개인 상황에서 FEM을 적용하기 위해 필요한 한주기 동안의  $\Delta t$ 의 개수는 최대 3개이며, 히터가  $m$ 개일 때는 최대  $m+1$ 개 이다. 본 논문에서는 그림 5의 히터 제어주기  $n$ 을 5초로 설정했다. 그림 7에는 제어기를 포함한 전체적인 시뮬레이션 구조를 나타내었다.

듀티 비를 사용자가 원하는 대로 조절하여 시뮬레이션 하고 원하는 지점의 온도 데이터를 얻음으로써 시스템 식별(system identification) 기법 등을 사용한 모델링 및 제어기 설계를 위한 기본 환경을 구축할 수 있었다.

[표 3] 시뮬레이션에 사용된 변수 및 값

변수	값	변수	값
길이	0.5 [m]	$\rho$	7800 [kg/m <sup>3</sup> ]
지름	0.15 [m]	$C$	460 [J/kg K]
물체 초기온도	293 [K]	$k$	45 [W/mK]
주변 공기온도	293 [K]	$h$	10 [W/m <sup>2</sup> K]



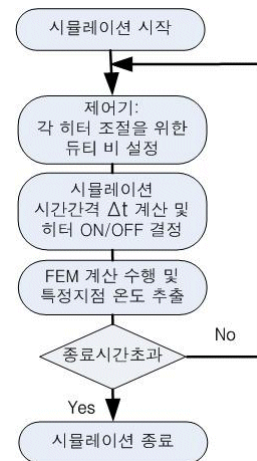
[그림 4] 외형 디자인 및 메쉬 생성



[그림 5] 히터 ON/OFF 동작



[그림 6] 듀티 비에 따른 시뮬레이션 시간 간격 및 히터 상태

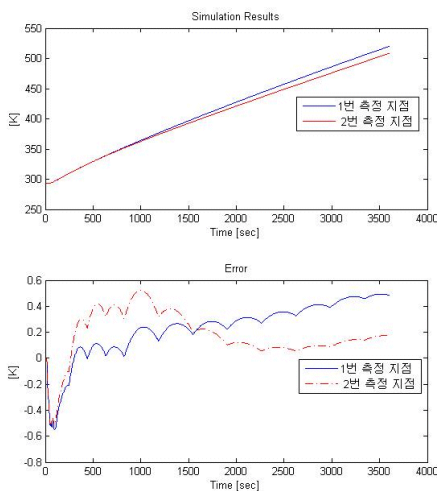


[그림 7] 제어기를 포함한 시뮬레이션 구조

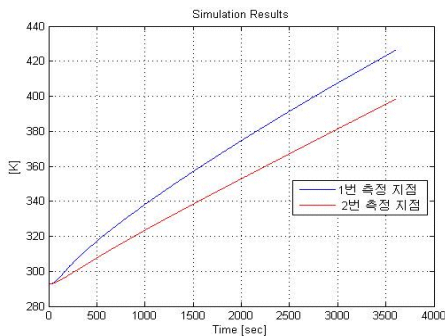
상용 FEM 소프트웨어인 FEMLAB [6]과 수치해석 결과를 간단히 비교해보기 위해서 히터 1번과 2번이 항상 켜져 있게 각각의 듀티 비를 1로 설정하고 3600초 동안 시뮬레이션 한 결과를 그림 8에 나타내었다. 이 때 온도 측정 지점은 그림 4에 표시했으며 1번 측정 지점은  $(x,y) = (0.15, 0.075)$ 이며 2번 측정 지점은  $(x,y) = (0.3, 0.075)$ 이다. 상용 FEM 소프트웨어와의 수치해석 결과 차이가 거의 없음을 알 수 있다.

다음은 히터 1번과 2번의 듀티 비가 각각 0.6과 0.2인 상황을 가정해 보자. 제안된 환경에서는 제어기에 의해서 듀티 비가 조절되는 상황을 손쉽게 구현 할 수 있으며 시뮬레이션 결과를 그림 9에 나타내었다.

그림 6에서 알 수 있듯이 2개의 히터 제어 시 한 주기 동안의 세부 FEM 시뮬레이션 시간 간격인  $\Delta t$ 의 개수는 최소 1개에서 최대 3개이다. 따라서 시뮬레이션에 걸리는 시간도 최대 3배 정도 차이가 날 것이다.



[그림 8] 시뮬레이션 결과비교 (모든 듀티 비: 1)



[그림 9] 시뮬레이션 결과  
(히터 1과 2의 듀티 비: 0.6과 0.2)

PC(CPU: 3GHz, RAM: 512MB)에서 표 4에 나타낸 바와 같이 히터 1과 2의 듀티 비가 모두 1인 경우와 각각 0.6과 0.2인 경우에 대해 3600초 까지 시뮬레이션 하는데 걸리는 시간을 측정해보니 대략 3배 정도 차이가 났으며 최대 시뮬레이션에 걸리는 시간이 대략 31초로 매우 적절한 수준이라 할 수 있다. 만약 복사 열전달과 온도에 따른 비열 및 열전도도의 비선형성을 고려하면 반복적인 해법(iteration)을 사용하게 되기 때문에 계산시간이 늘어날 수밖에 없지만 최대 반복 횟수를 적절히 제한하여 사용할 수 있다. 만약 MATLAB용 FEM 코드의 계산 속도가 문제가 된다면 [16]과 같은 FEM C++ 라이브러리를 이용하여 MATLAB과 연동되도록 디자인 할 수도 있다.

[표 4] 시뮬레이션 수행 시간

듀티 비		시뮬레이션 수행시간 [sec]
히터 1	히터 2	제안 환경
1	1	10.37
0.6	0.2	30.84

### 3. 결론

본 논문에서는 편리한 수학적 알고리즘 개발 환경이 장점인 MATLAB에서 온도 제어기 설계를 위한 2D FEM 시뮬레이션 환경 프로토타입을 구축했다. 외형 디자인 및 메쉬 생성, 제어기가 포함된 구조의 FEM 해석을 다루었으며 온도제어를 필요로 하는 다양한 산업 응용분야에서의 모델링 및 제어기 설계 과정에서 수행하게 되는 반복적인 열전달 실험에 비해 시간이 매우 단축되며 가열 후 재 실험을 위한 냉각 과정을 거칠 필요가 없어서 편리하다. 또한 사용자의 목적에 따라 코드 변형이 용이하며 MATLAB 코드에 직접적인 삽입이 가능하므로 다양한 상황에서의 온도 제어기 설계를 위한 연구 도구로 유용하게 응용될 수 있을 것이다. 향후 속도 향상을 위해 C언어를 이용하여 주요 부분을 구현하고 MATLAB과 연동되도록 구성되어야 할 것이다.

### 참고문헌

- [1] O. C. Zienkiewicz and R. L. Taylor, "The Finite Element Method, Volume 1: The Basis, fifth ed.", Butterworth-Heinemann, 2000.
- [2] R. W. Lewis, K. Morgan, H. R. Thomas, K. N. Seetharamu, The finite element method in heat

- transfer analysis, John Wiley & Sons Ltd, 1996.
- [3] Y. W. Kwon and H. C. bang, The finite element method using MATLAB 2nd ed., CRC press, 2000.
- [4] 장유진, “제어공학 분야에서의 유한요소법의 활용”, 전기학회논문지, 58권, 9호, pp. 1815-1820, 9월, 2009.
- [5] I. Nanno, N. Matsunaga, S. Kawaji, "A new model with feedback structure for thermal process and its application to decoupling control", IEEE Industrial Electronics, IECON 32nd Annual Conference, pp.507-512, 2006.
- [6] Introduction to FEMLAB product:  
[http:// comsol.com/products/](http://comsol.com/products/)
- [7] Structural Dynamics Toolbox, SDTools,  
<http://www.sdtools.com>
- [8] Third-Party Products & Services, MSC. visual Nastran 4D, [http://www.mathworks.com/products/connections/product\\_detail/product\\_35580.html](http://www.mathworks.com/products/connections/product_detail/product_35580.html)
- [9] W. Shijun and Z. Jinjuan, “FEM optimization for robot structure”, IEEE ICIT, pp. 510-513, 2002.
- [10] ANSYS APDL Programmer's Guide,  
[http://uic.edu/depts/accc/software/ansys/html/prog\\_55/g-apdl/AToc.htm](http://uic.edu/depts/accc/software/ansys/html/prog_55/g-apdl/AToc.htm)
- [11] S. Hölz, Polygon Clipper, 2005.  
<http://www.mathworks.com/matlabcentral/fileexchange/8818>
- [12] J. F. Pina, GTRACK, 2007.  
<http://www.mathworks.com/matlabcentral/fileexchange/15099>
- [13] P. -O. Persson, “Mesh Generation for Implicit Geometries”, Ph.D. Thesis, Department of Mathematics, MIT, 2004.
- [14] P. J. Frey and P. -L. George, Mesh Generation - Application to finite elements, WILEY, 2008.
- [15] S.S. Rao, The Finite Element Method in Engineering 4th ed., Elsevier Science & Technology Books, 2004.
- [16] Workbook of applications in VectorSpace C++ library, <http://www.vector-space.com/>

## 장 유 진(Yu Jin Jang)

[정회원]



- 1996년 2월 : 한국 과학기술 대학교 (공학사)
- 1998년 2월 : POSTECH (석사)
- 2002년 2월 : POSTECH (박사)
- 2000년 3월 ~ 2005년 2월 : 철강 제어 연구센터 연구원
- 2005년 3월 ~ 현재 : 동국대학교 정보통신공학부 조교수

<관심분야>

시스템 모델링, 비선형 시스템, 프로세스 자동화