



CUDA와 OpenMP를 이용한 NEC-2 코드 속도 향상

정종성*¹, 권혁준*², 최인식**³, 조용희***⁴, 오성원****⁵

Improvement of Calculation Speed of NEC-2 Code Using CUDA and OpenMP

Jong-Sung Jung*¹, Hyeok-Jun Kwon*², In-Sik Choi**³, Yong-Heui Cho***⁴, and Seong-Won Oh****⁵

본 연구는 국방과학연구소의 지원(계약번호:UD170002DD)을 받아 수행하였으며, 이에 감사드립니다.

요 약

전자파 수치 해석 기법 중 모멘트법(MoM : Method of Moment)에 기반한 NEC-2 코드는 대용량 행렬을 계산하는데 어려움이 있다. 본 논문에서는 계산 시간이 많이 걸리는 NEC-2 코드에 대하여 병렬처리 기법을 적용하여 계산 속도를 향상시켰다. NEC-2 코드의 모멘트 산란 행렬 계산부는 복잡한 제어 연산이 필요함에 따라 병렬코어 기반의 OpenMP(Open Multi-Processing)를 적용하였으며, NEC-2 코드의 가우스 소거법을 통해 계산되는 역행렬 연산에는 고속 병렬화가 가능하도록 하는 CUDA(Compute Unified Device Architecture)기반의 cuBLAS를 적용하였다. 시뮬레이션 결과 동일한 표적에 대하여 기존의 NEC-2코드와 계산 결과가 잘 일치하며, 해석 속도는 약 12배 이상 향상됨을 확인하였다.

Abstract

The NEC-2 code based on MoM(Method of Moment) among electromagnetic numerical analysis methods has difficulty in calculating a massive matrix. In this paper, we reduced the computation time by applying the parallel processing to the NEC-2 code, which needs a lot of computation time. The OpenMP(Open Multi-Processing) based on parallel cores is applied since the moment scattering matrix needs the complex operations. The cuBLAS based on CUDA(Compute Unified Device Architecture) which make the high speed parallelism possible, is applied to the inverse matrix calculation of NEC-2 code using the Gauss elimination. The simulation results show that the calculation results of the conventional NEC-2 code and our method are matched well. Furthermore, the calculation speed is faster about 12 times than the conventional NEC-2 code.

Keywords

OpenMP, CUDA, cuBLAS, NEC-2, radar cross section

* 한남대학교 무인시스템공학과

- ORCID¹: <http://orcid.org/0000-0002-2122-6434>

- ORCID²: <http://orcid.org/0000-0002-1432-2102>

** 한남대학교 전자공학과 교수(교신저자)

- ORCID: <http://orcid.org/0000-0002-8210-0843>

*** 목원대학교 정보통신융합공학부

- ORCID: <http://orcid.org/0000-0003-4618-8574>

**** 국방과학연구소 6본부 함정전투체계개발단

- ORCID: <http://orcid.org/0000-0001-5626-9891>

• Received: Nov. 30, 2017, Revised: Dec. 21, 2017, Accepted: Dec. 24, 2017

• Corresponding Author: In-Sik Choi

Dept. of Electronic Engineering, Hannam University, 70 Hannam-ro,

Daeduk-gu, Daejeon, 34430, Korea,

Tel.+82-42-629-7907, Email: recog@hnu.kr

I. 서 론

항공기나 선박과 같은 대형 표적에 대해서 실제 RCS(Radar Cross Section) 및 EMI(Electro Magnetic Interference) / EMC(Electro Magnetic Compatibility)를 측정 하는 것은 실내 측정 환경을 구축하는데 어려움이 있으며 실외 측정의 경우에도 다중경로 문제 등으로 인해 여러 가지 문제가 있다. 하지만 컴퓨터의 발전과 전파 해석 소프트웨어의 발전으로 RCS 및 EMI/EMC를 시뮬레이션을 통해 획득하는 것이 가능하게 되었다[1].

CPU(Central Processing Unit)의 발전은 싱글코어의 클럭 속도를 높이거나 성능을 향상시키는 방향이었다. 하지만 CPU 제조의 한계에 부딪침에 따라 싱글코어에서 코어 수를 늘리는 멀티코어 방식으로 발전하였다. 기존의 순차적인 프로그램은 싱글코어로 구현되어 있어 멀티코어를 가진 하드웨어라도 순차적인 코드의 성능은 향상되지 않았다. 따라서 순차적인 코드가 멀티코어에서 병렬화로 동작시키기 위해 OpenMP(Open Multi-Processing)가 개발되었고 프로그램의 병렬화를 위한 표준이 되었다[2].

GPU(Graphics Processing Unit)는 프로세서의 연산 효율을 향상시키는 방향으로 발전하게 되었다. GPU 코어를 대량으로 증가 시키고 빠른 메모리를 채택하여 GPU와 메모리 간 데이터 전송 격차를 줄였으며 대용량 데이터를 병렬처리 하는데 좋은 해결 방안이 되었다. CUDA(Compute Unified Device Architecture)는 강력해진 GPU의 성능을 이용하여 GPGPU의 통합 개발 환경을 제공한다. GPGPU는 ‘General Purpose Graphics Processing Unit’의 약자이며, 강력해진 GPU의 성능을 그래픽 처리뿐만 아니라 일반적인 데이터 처리에도 사용하는 것을 의미한다[3]-[5].

본 논문에서는 모멘트법 기반의 전파수치해석 코드인 NEC-2를 병렬코어 기반의 OpenMP와 CUDA 기반의 cuBLAS를 이용하여 병렬처리 알고리즘을 제안하였다. NEC-2는 포트란 언어를 기반으로 개발된 32bit 코드이다[6]. 32bit 코드로 개발된 NEC-2 코드는 OS(Operating System)에 의한 메모리 제한이 있어서 큰 구조물에 대하여 해석이 불가능하다. 따

라서 32bit 코드를 64bit 코드로 변경하여 메모리의 제한을 해결하고 전파 해석 하는데 시간이 많이 소요되는 부분에 대하여 병렬처리 기법을 적용하여 성능을 대폭 향상 시켰다. 또한 기존의 NEC-2 코드와 개발된 코드 그리고 상용화 툴인 FEKO를 이용하여 해석 결과를 검증하고 성능 비교를 하였다.

II. NEC-2 코드의 병렬처리 구현

2.1 OpenMP

여러 개의 코어로 이루어진 멀티 코어라도 순차적으로 작성된 프로그램의 코드는 성능 향상은 기대하기 어렵다. 하지만 순차적으로 작성된 프로그램에 특정 부분만을 병렬처리로 바꾸어 준다면 하드웨어적인 차이는 있지만, 크게 성능 향상이 된다. 이러한 문제를 해결하기 위해 개발된 OpenMP는 비교적 적은 노력으로 순차적 실행 프로그램을 병렬화하는데 효과적인 도구이다. OpenMP는 멀티프로세서 병렬 프로그래밍 모델을 위한 API(Application Programming Interface)이며, 프로그램의 병렬화를 위한 표준이다. 또한 여러 플랫폼에서 지원을 한다. 현재 많은 컴파일러에서 기본적으로 제공되고 있으며, OpenMP를 사용하기 위하여 설치하고 설정하는 과정을 생략할 수 있다. OpenMP는 직접 스레드를 생성하고 관리함에 따라 사용자가 쉽게 병렬처리 프로그램의 개발이 가능하다[2].

2.2 CUDA

GPU의 가속 컴퓨팅을 위한 CUDA는 NVIDIA에서 개발한 GPU 아키텍처이다. GPU는 CPU와 하드웨어적인 차이를 보이며, 현재 CPU는 강력한 순차 코드 성능을 지원하는 4개에서 16개 정도의 큰 프로세서 코어로 구성되어 있지만 GPU는 수천개에서 수만개의 작은 프로세서로 구성되어 있다[3]-[5].

2.3 NEC-2 코드의 병렬화

NEC-2 코드는 모멘트법을 이용함에 따라 모멘트

산란 행렬 계산이 필요하다. 모멘트 산란 행렬 계산 부는 다소 복잡한 제어 연산이 필요함에 따라 병렬 코어 기반의 OpenMP를 적용하였다. 행렬을 생성시에 OpenMP를 적용하여 성능을 향상시켰으며, 산란 행렬 생성은 원소별 독립 계산이 가능함으로써 매우 큰 와이어 매쉬에 대해서도 병렬 코어 개수에 비례하는 연산 가속화가 가능하다.

NEC-2에서 표면의 전류 분포 I 를 구하기 위한 행렬 연산은 다음과 같다.

$$ZI = V \rightarrow I = Z^{-1}V \quad (1)$$

이때 I 는 우리가 구하고자 하는 미지수인 I_m 을 원소로 갖는 벡터이며 V 는 주어진 전압 벡터이다.

이 행렬 계산은 가우스 소거법을 통해서 해결되는데 기본 단계는 가우스 소거법을 조금 변형시켜 LU분해를 적용하여 행렬 Z 를 식 (2)와 같이 상 삼각 행렬 U 와 하 삼각 행렬 L 의 곱으로 분해를 한다 [6]-[8].

$$Z = [L][U] \quad (2)$$

그러면 식 (1)은 식 (3)과 같이 된다[6][8].

$$[L][U][I] = [V] \quad (3)$$

행렬 I 를 구하기 위해서는 다음과 같이 두 단계로 계산된다[6][8].

$$[L][F] = [V] \quad (4)$$

식 (4)와 같이 전진대입법을 사용하여 $[F]$ 를 구한다[6][8].

$$[U][I] = [F] \quad (5)$$

식 (5)와 같이 역대입법을 사용하면 $[I]$ 에 대하여 풀리게 된다[8].

표면 전류 도출을 위한 가우스 소거법은 순차적

연산이 주를 이루기 때문에 OpenMP를 이용한 병렬화는 효율성이 떨어진다. 이에 따라 고속 병렬화를 구현하기 위해 CUDA 기반의 cuBLAS를 적용하여 NEC-2 코드의 해석 속도를 개선하였다. cuBLAS는 CUDA로 가속화 된 BLAS(Basic Linear Algebra Subroutines) 기능을 제공하는 가장 보편적인 라이브러리 중 하나이다[5]. BLAS는 벡터 및 행렬 연산의 문제를 효율적으로 구하기 위한 수학적 소프트웨어 패키지이다.

CUDA 기반의 cuBLAS는 LU분해를 위한 라이브러리를 제공하며 함수명은 zgetrf()와 zgetrs()으로 구현되어 있다. zgetrf()는 행렬의 LU 인수분해를 수행하기 위하여 사용되며, zgetrs()는 전진대입법과 역대입법을 해주는 루틴이다. cuBLAS를 이용한 LU분해의 코드 구현은 표 1과 같다[9].

표 1. cuBLAS를 이용한 LU분해의 코드
Table 1. Code of LU decomposition using cuBLAS

cuBLAS - LU decomposition code
<pre> INT_LAP nInfo; INT_LAP nM = (INT_LAP)n; INT_LAP nN = (INT_LAP)n; INT_LAP nLda = (INT_LAP)ndim; LibCuda::zgetrf(&nM, &nN, (COMPLEX_LAP *)a_in.data(), &nLda, (INT_LAP *)ip.data(), &nInfo); </pre>
cuBLAS - Matrix calculation using Gauss elimination
<pre> INT_LAP nInfo; CHAR_LAP trans = 'N'; INT_LAP nN = n; INT_LAP nNrhs = 1; INT_LAP nLda = (INT_LAP)ndim; INT_LAP nLdb = n; LibCuda::zgetrs(&trans, &nN, &nNrhs, (COMPLEX_LAP *)a.data(), &nLda, (INT_LAP *)ip.data(), (COMPLEX_LAP *)b.data(), &nLdb, &nInfo); </pre>

그림 1은 NEC-2 코드 병렬화 과정을 보여준다. 구조물에 대한 매쉬의 입력이 들어오면 OpenMP를 통해 코어 수만큼 분해하여 들어가고 행렬을 생성 후 표면 전류를 도출하기 위한 행렬 연산은 cuBLAS 기반의 BLAS를 통하여 계산된다.

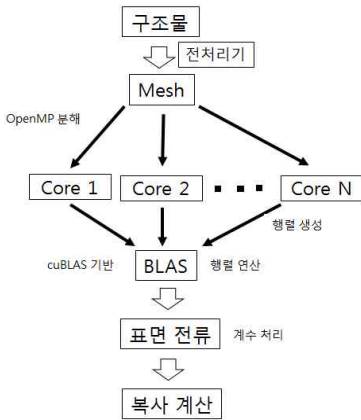


그림 1. NEC-2 코드 병렬화
Fig. 1. Parallelizing for NEC-2 code

III. 시뮬레이션 결과 및 분석

기존의 NEC-2코드와 병렬처리를 구현한 NEC-2 (OpenMP+CUDA)코드 그리고 상용화 해석 툴인 FEKO를 사용하여 같은 표적에 대한 해석 결과를 검증하고 해석 시간 및 메모리 사용량 비교하였다.

3.1 실험 환경

표적에 대한 검증 및 해석 시간 측정을 위한 컴퓨터와 그래픽 카드의 구성은 표 2와 표 3과 같다.

표 2. 시뮬레이션 컴퓨터 환경
Table 2. Environment of simulation computer

	Simulation environment
CPU	Intel Xeon(R) CPU E5-2670 2.6GHz (2개)
Memory	DDR3 128GB
GPU	NVIDIA GeForce GTX 650 Ti
OS	Windows 7 Professional K 64bit
Development Tool	Visual Studio 2013
EM Software	FEKO 7.0

표 3. 그래픽 카드의 환경
Table 3. Environment of graphic card

	Simulation environment
CUDA Core	768
Engine Clock	978MHz
Memory	GDDR5 1GB
Memory Clock	5400MHz

3.2 RCS 정확도 검증

RCS에 대한 정확도 검증을 위하여 와이어 메쉬를 가지는 구를 모델링하였다[10]. 구에 대한 RCS는 정확한 이론적인 해를 구할 수 있으며, 상용화 툴인 FEKO를 이용한 구의 RCS 값은 이론적인 해와 잘 일치한다[10]. 따라서 상용화 해석 툴인 FEKO와 기존의 NEC-2코드, openMP와 CUDA 기반의 cuBLAS를 적용한 NEC-2(OpenMP+CUDA)코드를 사용하여 구에 대한 RCS를 해석하고 정확도 및 계산속도를 검증하였다[11]. 모델링한 구는 그림 2와 같다.

실험에 사용된 주파수는 VHF대역으로[11] 30MHz에서 300MHz까지 55포인트 간격이며, 구의 반지름은 0.5m이다. 메쉬를 나누는 기준은 300MHz를 기준으로 $\lambda/20$ 로 계산하였으며 와이어의 총 개수는 4950개이며, 삼각형 총 개수는 4022개이다.

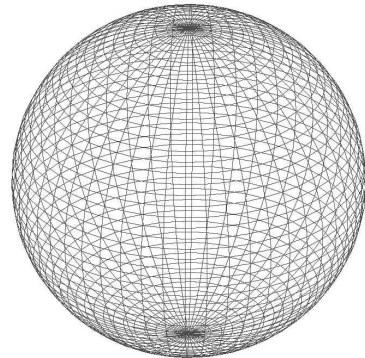


그림 2. 와이어 메쉬로 모델링한 구
Fig. 2. Sphere modeled by wire mesh

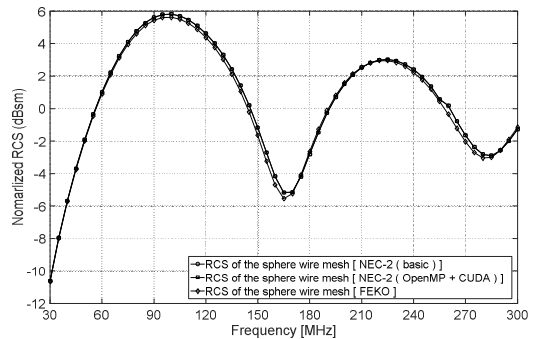


그림 3. 와이어 메쉬 구의 RCS
Fig. 3. RCS of the sphere wire mesh

표 4. 4950개 와이어 메쉬 구의 RCS 해석 시 계산 시간 및 메모리 사용량

Table 4. Calculation time and memory usage for RCS analysis of 4950 wire mesh sphere

	Time	Memory
NEC-2(Basic)	151 min. 10 sec.	0.46[Gbyte]
NEC-2(OpenMP+CUDA)	8 min. 27 sec.	0.91[Gbyte]
FEKO(MoM)	30 min. 37 sec.	0.49[Gbyte]

3.3 Far-field와 Near-field 정확도 검증

Far-field의 정확도 검증을 위하여 와이어 메쉬를 가지는 구를 모델링하고 다이폴 안테나를 와이어 메쉬로 모델링하였으며, 다이폴 안테나에 전원을 공급하고 다이폴 안테나의 방사된 전자파가 구에 미치는 영향을 확인하기 위하여 Far-field 영역에 대하여 방사 패턴을 통해 정확도를 검증하였다.

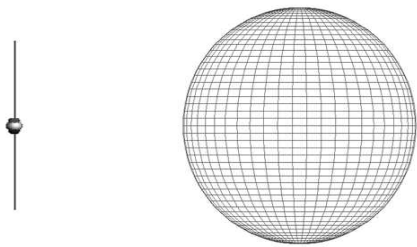


그림 4. 와이어 메쉬로 모델링한 구와 다이폴 안테나
Fig. 4. Sphere and dipole antenna modeled by wire mesh

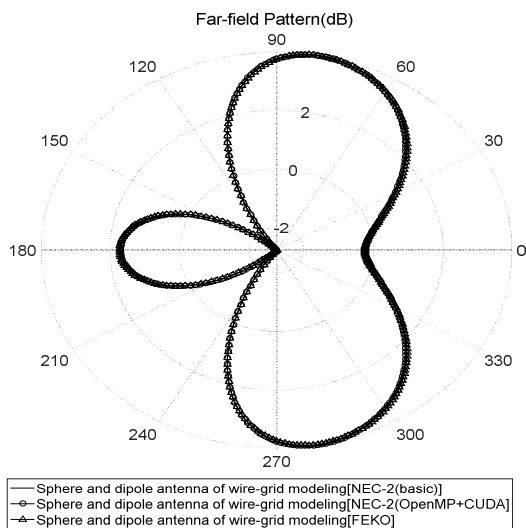


그림 5. 그림 4에 대한 원전계 패턴
Fig. 5. Far-field pattern of Fig. 4

구의 반지름은 0.5m이고 구의 중심은 (0, 0, 0)이다. 다이폴 안테나 길이는 (1.2, 0, -0.375)에서 (1.2, 0, 0.375)까지 0.75m이며, 반지름 0.0001m이다.

Near-field는 다이폴 안테나에 전원을 공급하고 방사되는 전자파가 구로부터 산란되는 결과를 통해 검증하였으며, 구의 중심(0, 0, 0)을 기준으로 구 좌표계를 사용하여 $r=0.6m$, $\theta=90^\circ$, $\phi=0 \sim 360^\circ$ 로 1° 간격으로 측정하였다. 측정 위치는 그림 6과 같다.

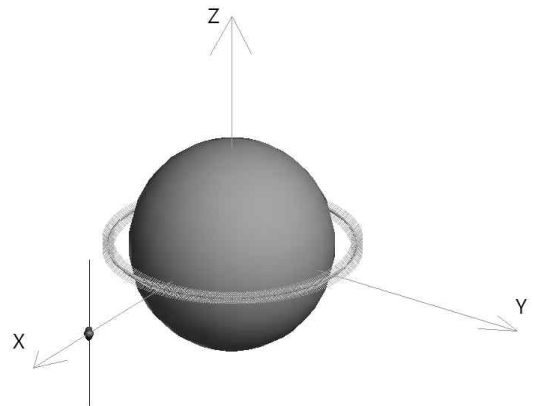


그림 6. Near-field 측정 위치
Fig. 6. Near-field measurement point

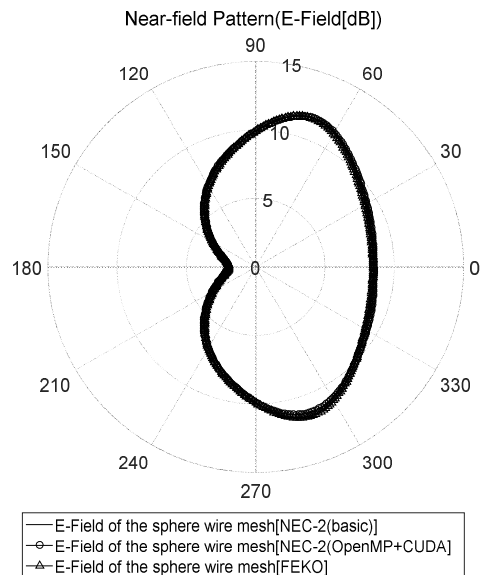


그림 7. 그림 6에서 구의 표면으로부터 0.1m 떨어진 곳에서의 Near-field 값
Fig. 7. Near-field pattern of Fig. 6 when the distance is 0.1m from sphere surface

표 5. Far-field와 Near-Field의 계산 시간 및 메모리 사용량
Table 5. Calculation time and memory usage of Far-field and Near-Field cases

	Time	Memory
NEC-2(Basic)	2 min. 42 sec.	0.46[Gbyte]
NEC-2(OpenMP+CUDA)	14 sec.	0.92[Gbyte]
FEKO(MoM)	35 sec.	0.49[Gbyte]

IV. 결 론

본 논문에서는 대용량 행렬 연산이 필요한 모멘트법 기반의 기존의 NEC-2코드에 OpenMP와 CUDA 기반의 cuBLAS를 프로그래밍 하여 성능을 향상시켰다. NEC-2 코드는 모멘트법을 이용함에 따라 모멘트 산란 행렬 계산이 필요하다. 모멘트 산란 행렬 계산부는 다소 복잡한 제어 연산이 필요함에 따라 병렬코어 기반의 OpenMP를 적용하였으며, 또한 NEC-2코드의 행렬 연산은 가우스 소거법을 통해 계산되는데, 가장 많은 시간이 걸리는 부분이다. 시간이 많이 걸리는 행렬 연산의 가우스 소거법을 고속 병렬화가 가능하도록 하는 CUDA기반의 cuBLAS를 구현하여 성능을 향상시켰다. 성능 향상이 된 NEC-2(OpenMP+CUDA) 코드는 상용화 툴인 FEKO와 RCS/Far-field/Near-Field의 계산 결과를 비교하였을 때, 매우 잘 일치하는 것을 확인 하였다. 또한 NEC-2(OpenMP+CUDA) 코드는 NEC-2코드 보다 약 12배, FEKO보다 몇배 이상 해석 속도가 향상이 되었다.

다양한 해석 기법에도 병렬처리기법을 적용하면 성능이 향상 할 것으로 예상됨에 따라 추가적인 연구가 진행되어야 한다.

References

- [1] S. W. Oh, "The Performance Analysis of Wire Grid and RWG Model in MoM", Postech Master's Thesis, 2003.
- [2] R. Chandra, "Parallel programming in OpenMP", Morgan kaufmann, 2001.
- [3] Kirk, David B. and Wen-mei W. Hwu, "Programming Massively Parallel Processors: A

Hands-on Approach", Morgan Kaufmann Publishers, 2010.

- [4] Y. K. Ko, "Designing and Implementing parallel processing software for high-speed calculating process based on CUDA", Chungnam National University Master's Thesis, 2013.
- [5] Y. H. Jung, "CUDA parallel programming - NVIDIA parallel computing architecture CUDA using high performance GPGPU", FREELEC, 2011
- [6] G. J. Burke and A. J. Poggio, "Numerical Electromagnetics Code (NEC) Method of Moments Part II: Program Description-Code", Lawrence Livermore Laboratory, Jan. 1981.
- [7] E. Lezar and D. B. Davidson, "GPU-based LU decomposition for large method of moments problems", Electronics letters, Vol. 46, No. 17, pp. 1194-1196, Sep. 2010.
- [8] Gibson, Walton C., "The method of moments in electromagnetics", CRC press, 2014.
- [9] NVIDIA, Cublas library - User Guide, NVIDIA Corporation, 2017.
- [10] J. S. Jung, H. J. Kwon, and I. S. Choi, "A Study on the Analysis of Sphere RCS Using Time - Frequency Analysis", Journal of KIIT, Vol. 15, No. 4, pp. 89-94, Apr. 2017.
- [11] S. H. Im, S. J. Lee, and I. S. Choi, "A Study on Calculation and Comparison for VHF, UHF, L-Band RCS", Journal of KIIT, Vol. 14, No. 5, pp. 59-66, May 2016.

저자소개

정 종 성 (Jong-Sung Jung)



2016년 2월 : 한남대학교
전자공학과(공학사)
2016년 3월 ~ 현재 : 한남대학교
무인시스템공학과 석사과정
관심분야 : EMI/EMC, 전자파
수치해석

권혁준 (Hyeok-Jun Kwon)



2016년 2월 : 한남대학교
전자공학과(공학사)
2016년 3월 ~ 현재 : 한남대학교
무인시스템공학과 석사과정
관심분야 : EMI/EMC, 전자파
수치해석

최인식 (In-Sik Choi)



1998년 2월 : 경북대학교
전자공학과(공학사)
2000년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2003년 2월 : 포항공과대학교
전자전기공학과(공학박사)
2004년 ~ 2007년 : 국방과학연구소

선임연구원

2007년 ~ 현재 : 한남대학교 전자공학과 교수
관심분야 : 레이더 신호처리, 레이더 시스템 설계, RCS
해석

조용희 (Yong-Heui Cho)



1998년 2월 : 경북대학교
전자공학과(공학사)
2000년 2월 : 한국과학기술원
전기및전자공학과(공학석사)
2002년 8월 : 한국과학기술원
전자전산학과 전기 및
전자공학전공(공학박사)

2002년 4월 ~ 현재 : 한국전자통신연구원(ETRI)
무선방송연구소 위촉·선임·초빙연구원
관심분야 : 안테나, 초고주파, 전자파 산란 및 분산

오성원 (Seong-Won Oh)



1996년 3월 : 해군사관학교
전기공학과(공학사)
2004년 2월 : 포항공과대학교
정보통신공학과(공학석사)
2011년 9월 : Texas A&M Univ.
전자공학과(공학박사)