

SDR 시스템에서 GPU를 사용한 Lattice Reduction-aided 검출기 구현*

김 태 현** · 이 현 식*** · 최 승 원****

Implementation of Lattice Reduction-aided Detector using GPU on SDR System

Kim Tae Hyun · Lee Hyun Seok · Choi Seung Won

〈Abstract〉

This paper presents an implementation of Lattice Reduction (LR)-aided detector for Multiple-Input Multiple-Output (MIMO) system using Graphics Processing Unit (GPU). GPU is a parallel processor which has a number of Arithmetic Logic Units (ALUs), thus, it can minimize the operation time of LR algorithm through the parallelization using multiple threads in the GPU. Through the implemented LR-aided detector, we verify that the LR-aided detector operates a lot faster than Maximum Likelihood (ML) detector. The implemented LR-aided detector has been applied to WiMAX system to show the feasibility of its real-time processing. In addition, we demonstrate that the processing time can be reduced at the cost of 3dB SNR loss by limiting the repeating loop in Lenstra-Lenstra-Lovasz (LLL) algorithm which is frequently used in LR-aided detector.

Key Words : Lattice Reduction, Graphics Processing Unit, MIMO Detector, Parallel Processing, SDR

I. 서론

최근 주파수 부족 문제가 대두되면서 CR(Cognitive Radio) 기술에 대한 관심이 높아지고 있다. CR 기술은 다양한 통신 기술과 주파수 사이를 필요에 따라 넘나들게 해주는 기술이고, 이 기술을 현실화시켜주는 것이 바로 SDR(Software Defined Radio) 기술을 이용하는 reconfigurable 단말기이다. 그리하여 앞으로 다가올 미래의 SDR 통신단말기를 위해 ETSI(European

Telecommunications Standards Institute)에서는 표준화가 진행되고 있다[1]. 이 표준을 만드는 과정에는 SDR 통신단말기의 검증과정이 필수적이다. 본 논문은 SDR 단말기가 실제 사용이 가능한지 검증하기 위해 GPU를 사용하여 이를 수행하였다.

최근 차세대 통신시스템에서는 높은 데이터 전송률을 보장하기위해 공간 다중화 방식 MIMO 기술의 필요성이 대두되고 있다. 이는 MIMO 시스템이 추가적인 주파수나 높은 송신전력의 지원 없이 데이터 전송량을 선형적으로 증가시킬 수 있기 때문이다[2]. 이러한 MIMO 시스템에서는 수신신호를 검출하기 위해 여러 종류의 검출기를 사용하는데 그 중에서 MLD(Maximum Likelihood Detector)는 최대 다이버시티 이득을 얻어 가장 이상적인

* 본 연구는 지식경제부의 지원을 받는 정보통신표준화 및 인증 지원사업의 연구결과로 수행되었음

** 한양대학교 전자컴퓨터통신공학과 통신신호처리연구실

*** 삼성전자 무선사업부 SW Platform팀 Assistant Engineer

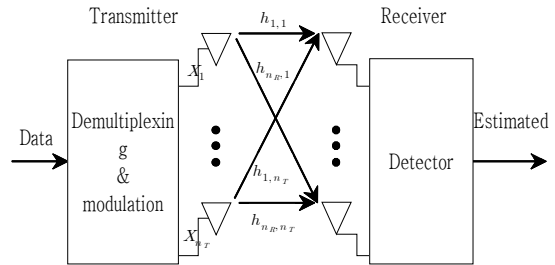
**** 한양대학교 전자컴퓨터통신공학과 교수(교신저자)

성능을 보여준다. 그러나 이 방법은 송신 안테나의 개수와 변조 크기가 높아짐에 따라 계산량이 지수적으로 증가하는 단점이 있다[3-4]. 이런 단점을 보완하기 위해 복잡도가 낮은 ZF(Zero-Forcing) 또는 MMSE(Minimum Mean Square Error)와 같은 검출 기법이[5-6] 제안되었으나 이런 방식들은 ML 기법에 비해 성능 열화가 심각하다. 이에 최근에는 LR(Lattice Reduction)을 이용하여 적은 계산량을 가지는 검출기법이 제안되었으며, LR을 사용한 검출기는 MLD에 가까운 성능을 보여준다[7].

LR을 사용한 검출기에서는 격자 감소를 위해 LLL(Lenstra-Lenstra-Lovasz) 알고리즘이 사용되고 있다 [8]. LLL 알고리즘은 우선 Gram-Schmidt 직교화 과정을 거친 뒤 열 벡터의 교환 과정이 반복적으로 수행되어 여전히 복잡도가 높고, 기존의 통신 시스템에서 쓰이는 DSP(Digital Signal Processor)과 같은 직렬 프로세서를 사용하면 검출기의 연산시간이 오래 걸린다.

본 논문에서는 LRAD(LR-Aided Detector)의 연산시간을 최소화하기 위해 GPU(Graphics Processing Unit)를 사용하여 LRAD를 구현하는 것을 제안한다. GPU는 다수의 thread로 구성되어 병렬 연산이 가능한 프로세서로 [9] 이를 사용하여 LR 알고리즘이 여러 thread에서 동시에 연산되도록 하여 연산시간을 줄이고자 한다. 또한 LRAD를 WiMAX 시스템에 적용시켜 LRAD가 MLD보다 더 빠르게 연산 처리되며 실시간 처리(real-time processing)가 가능한지를 확인하고 반복문 제한을 통해 더 빠른 연산도 가능하다는 것을 확인한다.

본 논문의 구성은 우선 II 장에서 시스템 구성과 기본적인 LR 알고리즘 및 GPU에 대해 설명하고 III장에서는 LRAD를 GPU에 구현하고 반복문을 제한에 대해 설명할 것이다. IV 장에서는 모의실험과 실제 WiMAX 시스템을 통해 구현된 시스템의 성능을 보이고 마지막으로 V 장에서 결론을 맺는다.



<그림 1> MIMO 시스템 모델
(송신 안테나 n_T 개, 수신안테나 n_R 개)

II. 시스템 모델

2.1 MIMO 시스템 구성

본 논문에서는 <그림 1>에 나타난 것처럼 n_T 개의 송신 안테나와 n_R 개의 수신 안테나를 사용하였다. 데이터 비트들은 역 다중화를 통해 n_T 개의 스트림으로 나뉘어서 변조되어(이 시스템에서는 16QAM 변조 방식을 사용하였다.) 송신 신호 벡터 $x = [x_1, \dots, x_{n_T}]^T$ 가 된다. 수신기에서는 각 안테나에서 수신된 수신 신호 벡터 $y = [y_1, \dots, y_{n_R}]^T$ 를 이용하여 전송된 데이터 스트림을 복원하게 된다. $[\]^T$ 는 트랜스포즈(transpose)를 뜻한다. 송신신호 벡터와 수신 신호 벡터와의 관계는 다음과 같다.

$$y = hx + n \quad (1)$$

여기서 $n = [n_1, \dots, n_{n_R}]^T$ 은 σ_n^2 의 분산을 가진 가우시안 잡음 벡터, h 는 $n_R \times n_T$ 로 구성된 채널 행렬을 나타낸다. 채널 행렬 h 는 flat fading 환경이고 이 채널은 완벽히 추정되는 걸로 가정한다.

위의 수식을 실수 모델로 다루기 위하여 시스템 모델을 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} R(y) \\ I(y) \end{bmatrix} = \begin{bmatrix} R(h) - I(h) \\ I(h) \quad R(h) \end{bmatrix} \begin{bmatrix} R(x) \\ I(x) \end{bmatrix} + \begin{bmatrix} R(n) \\ I(n) \end{bmatrix} \quad (2)$$

여기서 $R(\)$ 는 실수부분, $I(\)$ 는 허수부분을 나타낸

다. 이를 다시 나타내면 다음과 같다.

$$Y = HX + N \quad (3)$$

여기서 $H = \begin{bmatrix} R(h) - I(h) \\ I(h) \quad R(h) \end{bmatrix}$ 는 실수로 이루어진 $N_r \times N_t$ 행렬이다. ($N_t = 2n_T, N_r = 2n_R$)

2.2 Lattice Reduction

격자 $L(H)$ 는 기저 벡터(basis vector) h_i ($1 \leq i \leq N_R$)로 정의하고 h_i 는 실수로만 이루어진 채널 행렬 H 의 열이다. $N_t \times N_t$ 행렬 T 가 uni-modular 일 때 $\tilde{H} = HT$ 는 H 와 같은 격자를 가진다. 즉, 여기서 T 는 정수로 이루어져 있고 행렬식 $\det(T)$ 이 ± 1 이다. Lattice Reduction은 T 를 이용하여 기존의 행렬 H 를 새로운 행렬 \tilde{H} 로 바꾸어 주는 기법이다. 이를 통해 새로운 \tilde{H} 의 기저 벡터 \tilde{h}_i 들이 유사 직교성을 띄게 만든다.

2.2.1 LLL(Lenstra-Lenstra-Lovasz) algorithm

LLL 알고리즘은 크기 감소(size reduction)와 열 교환(column exchange)의 두 가지 연산으로 이루어져 있다.

$$|\mu_{i,j}| \leq 1/2 \quad \text{for } 1 \leq j < i \leq N_t \quad (4)$$

$$\mu_{i,j} = \frac{(h_i, h_j^*)}{\|h_j^*\|^2} \quad (5)$$

$$\|h_i^* + \mu_{i,i-1}h_{i-1}^*\| \geq \lambda \|h_{i-1}^*\|^2 \quad \text{for } 1 < i \leq N_t \quad (6)$$

여기서 $(h_i^*, \dots, h_{N_t}^*)$ 는 H 를 Gram-Schmidt 과정으로부터 구한 기저 벡터들이고, 매개변수 $\lambda(1/4 < \lambda \leq 1)$ 는 본문에서 3/4로 가정한다.

크기 감소에서는 $1 \leq i \leq N_t$ 를 만족하는 모든 i 에 대해 $|\mu_{i,j}| > 1/2$ 와 $j < i$ 를 만족하는 모든 열 벡터 h_j 를 구한다. 그 다음 아래 연산을 수행한다.

$$h_i \leftarrow h_i - I_{i,j}h_j \quad (7)$$

여기서 $I_{i,j}$ 는 $\mu_{i,j}$ 에 가장 가까운 정수이고 $\mu_{i,j}$ 값은 계속 업데이트된다.

두 번째로 열 교환 과정에서는 LLL 알고리즘이 두 개의 인접한 기저벡터를 서로 교환함으로써 ($h_i \leftrightarrow h_{i-1}$) 식 (6)의 조건을 만족하면서 직교성이 뛰어난 열 벡터를 찾는다. 위의 두 과정은 조건식 (4), (6) 모두를 만족시킬 때까지 수행된다.

2.2.2 LR-Aided Detector

$\tilde{H} = HT$ 이고 $Z = T^{-1}X$ 라 할 때 수신신호 벡터는 아래와 같다.

$$Y = HX + N = HTT^{-1}X + N = \tilde{H}Z + N \quad (8)$$

여기서 HX 와 $\tilde{H}Z$ 는 격자에서 같은 지점을 가리키나 LLL 과정에 의해 생성된 \tilde{H} 는 훨씬 더 좋은 조건의 행렬이다. 수신신호를 검출하는 식은 다음과 같다.

$$\tilde{X} = H^+ Y = X + (H^T H)^{-1} H^T N \quad (9)$$

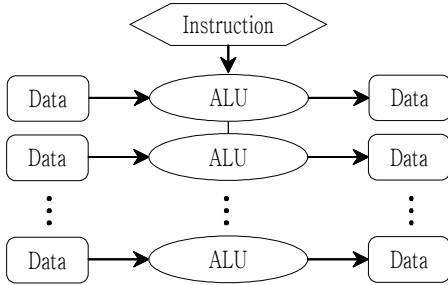
$$\tilde{Z} = T^{-1} \tilde{X} = \tilde{H}^+ Y = Z + \tilde{H}^+ N \quad (10)$$

여기서 $H^+ = (H^T H)^{-1} H^T$ 는 pseudo-inverse 과정을 거친 채널 행렬을 나타낸다. \tilde{H} 의 유사 직교성 때문에 \tilde{H}^+ 을 잡음에 곱하게 되면 H^+ 보다 더 적게 잡음을 키우게 된다.

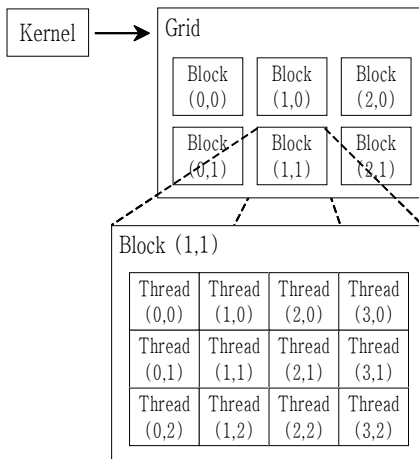
2.3 Overview of GPU

GPU는 SIMD(Single Instruction Multiple Data) 구조로서 병렬 처리에 뛰어난 성능을 보여주는 프로세서이다. <그림 2>에서와 같이 하나의 명령이 다수의 ALU(Arithmetic Logic Unit)를 사용하여 동시에 여러 데이터를 처리하게 된다[10].

<그림 3>에서는 thread와 block 구조를 나타내고 있는데 하나의 block안에 다수의 thread가 들어있고 다수의 block이 전체 구조를 형성한다. 여기에 속해있는 여러



<그림 2> SIMD 구조[10]



<그림 3> Grid of Thread Blocks[11]

thread에서 동시에 실행되는 함수를 kernel이라고 부른다[11]. 이처럼 하나의 명령이 주어지면 수많은 thread에서 동시에 커널이 실행되는 병렬 처리 연산으로 그 연산 시간을 획기적으로 줄일 수 있다.

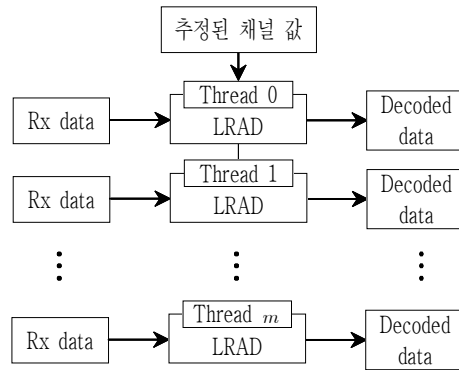
III. 구현

3.1 LRAD 구현

LLL 알고리즘은 2.2.1에서 설명하였듯이 크기 감소와 열 교환의 두가지 과정으로 이루어진다. 그리고 두 번째 과정인 열 교환에서는 조건식 (6)을 만족할 때까지 교환 과정을 수행한다.

<표 1> LLL 알고리즘

- (1) $k=2$
- (2) while $k \leq N_t$
- (3) size reduction
- (4) column exchange
- (5) update k
- (6) end



<그림 4> GPU에 구현된 LRAD 구조

<표 1>에 간단하게 설명된 LLL 알고리즘을 보면 알고리즘이 while문 안에서 돌아간다는 것을 알 수 있다. 이렇게 되풀이되며 돌아가는 while문은 직렬 처리(serial processing)를 할 수 밖에 없다.

그리하여 LRAD의 LLL 알고리즘은 다수의 thread에서 병렬 처리되도록 구현하였다. <그림 4>은 구현된 LRAD의 구조를 나타낸다. 수신 신호를 병렬화하여 $m+1$ 개의 thread에 입력하면 LRAD가 동시에 데이터를 연산 처리하여 디코딩된 데이터를 다음 단계로 전해 준다. 여기서 thread의 총 개수는 이후 2x2 MIMO WiMAX에 실험해보기 위해 17,280개로 맞췄다.

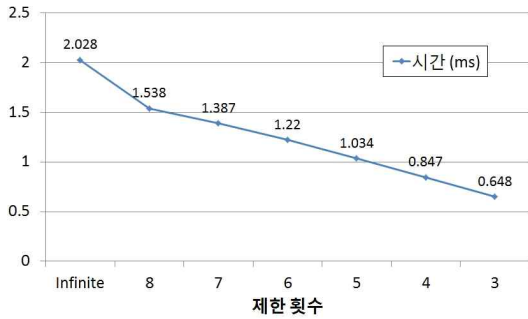
3.2 Iteration 제한

<표 1>에서 알 수 있듯이 LLL 알고리즘은 while문 안에서 반복적으로 수행되는 알고리즘이다. 이는 병렬 처리가 힘들기 때문에 알고리즘 내에서는 연산 시간을 줄

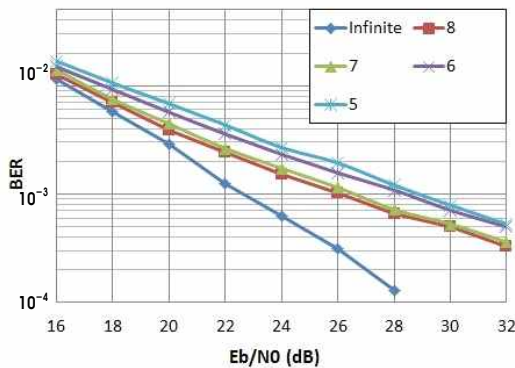
이는 데에 한계가 있다. 본 논문에서는 <표 1>의 while 문 안에서 알고리즘의 반복 횟수를 제한함으로써 더 빠른 연산처리가 가능하도록 구현하였다.

IV. 실험 결과

본 논문에서는 LR을 사용한 검출기를 GPU를 사용하여 구현하였을 때의 성능을 검증한다. 이를 위하여 2x2 MIMO 시스템에서 16-QAM 변조기법을 사용하였고, WiMAX 파형(waveform)에 맞춰 ML과 ZF 알고리즘을 사용한 검출기를 사용하여 비교 실험을 하였다. 여기서 채널은 flat fading 환경이고 <그림 6>과 <그림 7>에서 나오는 BER 그래프는 uncoded BER을 나타낸다.



<그림 5> while loop의 횟수 제한에 따른 연산 속도 변화



<그림 6> while loop 횟수 제한에 따른 BER 성능 ($n_T = n_R = 2$, 16QAM, uncoded)

4.1 While문 제한 실험 결과

컴퓨터 시뮬레이션을 통해 2x2 MIMO 시스템에서 LRAD의 while문이 평균적으로 약 7.11번 반복되는 것을 확인하였다. 그리하여 while 반복문에 제한을 두어 그에 따른 연산 속도의 변화를 측정하여 <그림 5>와 같은 그래프를 얻을 수 있었다. 그래프를 보면 기존 알고리즘에서는 LRAD가 2.028ms의 연산시간을 가지나, while문을 8번 까지만 반복되도록 하면 1.538ms로 줄어들고 그 횟수를 줄일수록 연산시간이 선형적으로 줄어드는 것을 볼 수 있다.

<그림 6>에서는 while문의 반복 횟수를 제한했을 때 그에 따른 BER 성능을 보여준다. while문 제한을 8번까지만 하면 성능이 3dB 안 좋아지지만 속도가 0.5ms 가량 빨라진다. <그림 5>와 <그림 6>에서 종합적으로 보면 7번으로 제한하여도 8번 제한했을 때와 같은 성능을 보이면서 연산 속도는 0.64ms정도 빨라지는 것을 확인할 수 있다. 이는 구현에 있어 상당한 개선으로 LRAD를 구현할 때 while loop를 7번까지만 반복하도록 제한을 두면 약간의 성능 열화를 감수하고 속도를 줄일 수 있다.

<표 2> WiMAX 시스템에서 ML, LR, ZF 알고리즘의 연산 속도 비교

Algorithm	ML	LR	ZF
MIMO detector	3.84ms	2.03ms	0.04ms
all blocks except detector in the receiver	2.33ms		
Total	6.17ms	4.36ms	2.37ms

4.2 연산 속도

<표 2>에서는 LR을 사용한 검출기의 연산 속도가 GPU의 병렬 처리 능력 덕분에 ML보다 배 가까이 빠른 것을 볼 수 있다. <그림 7>의 성능 그래프를 보면 LRAD

가 MLD보다 훨씬 빠르면서 이상적인 BER 성능을 보여주는 ML에 비해 크게 떨어지지 않는 성능을 보여준다.

<표 2>의 실험에 쓰인 WiMAX 시스템은 수신신호가 2x2 MIMO detector에 들어가기 전 한 프레임에 17, 280 개의 변조된 신호들이 있다. 업링크를 사용하지 않을 시에 실시간 연산이 가능하려면 MIMO 검출기를 포함하여 수신단에서 걸리는 시간이 5ms보다 작아야 한다. <표 2>의 마지막 행을 보면 수신 단에서 총 걸린 시간이 나와 있는데 검출기에 ML 알고리즘을 쓴 경우에는 6.17ms 이나 걸려 실시간 연산이 불가능하나 LRAD를 사용함 경우에 총 걸린 시간은 4.36ms로 실시간 연산이 가능함을 보여준다.

V. 결론

본 논문에서는 LRAD를 구현함에 있어 기존의 직렬 프로세서 대신 병렬 처리가 가능한 GPU를 사용하는 것을 제안하였다. 제안하는 구현 시스템은 LR 알고리즘이 GPU의 여러 thread에서 동시에 연산 처리되도록 하여 full diversity를 얻는 ML 알고리즘보다는 약 3dB의 손실을 보이지만 그 연산시간은 거의 두배 가량 빠르다는 것을 보여주었다. 또한, while문 제한으로 필요에 따라 더욱 빠르게 연산도 가능함을 보여주었고, WiMAX 시스템에 적용시켜 실시간 연산도 가능함을 확인하였다. 따라서 본 논문은 향후 차세대 통신기술을 이용한 기기에서 LRAD를 GPU를 사용해서 구현할 수 있다는 가능성을 제시하였다.

참고문헌

- [1] M. Mueck, A. Piipponen, K. Kalliojarvi, G. Dimitrakopoulos, K. Tsagkaris, and P. Demestichas, "ETSI Reconfigurable Radio Systems: Status and Future Directions on Software Defined Radio and Cognitive Radio Standards," *IEEE Communications Magazine*, SEP 2010, pp. 78-86.
- [2] S. Haykin and M. Moher, *Modern Wireless Communications*, Pearson Prentice Hall, New Jersey, 2005.
- [3] S. Verdú, *Multiuser Detection*. Cambridge, U. K: Cambridge Univ. Press, 1998.
- [4] 홍경화 · 오태열 · 최승원, "WiBro 다운로드 환경에서 여러 가지 MIMO 기법의 성능 및 연산량에 대한 비교 연구," *디지털산업정보학회, 디지털산업정보학회논문지*, 제 4권 제 2호, 2008.
- [5] John Proakis, *Digital communications*, 4th ed., New York, NY, McGraw- Hill, 2000.
- [6] Drink Wübben, Ronald Böhnke, Volker Kühn, and Karl-Dirk Kammeyer, "Near-Maximum-Likelihood Detection of MIMO Systems using MMSE-based Lattice-Reduction," *2004 IEEE International Conference on Communications*, Vol. 2, 2004, pp. 798-802.
- [7] Huan Yao and W. Wornell, "Lattice-Reduction-Aided Detectors for MIMO Communication Systems," *Global Telecommunication Conference*, Vol. 1, 2002, pp. 424-428.
- [8] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring Polynomials with Rational Coefficients," *Math. Ann.*, Vol. 261, 1982, pp. 515-534.
- [9] June Kim, Seungheon Hyeon, and Seungwon Choi, "Implementation of an SDR System Using Graphics Processing Unit", *Communications Magazine of the IEEE*, Vol. 48, 2010.
- [10] Jaehyuk Ju, Chiyoung Ahn, June Kim, Seungheon Hyeon, and Seungwon Choi, "Implementation of an SDR platform using GPU and its application to 2x2 MIMO WiMax system", *Wireless*

Innovation Forum, 2010, pp. 341-345.
[11] NVIDIA Corporation, CUDA C Programming Guide, 2010.

논문접수일 : 2011년 4월 15일
수정일 : 2011년 5월 9일
게재확정일 : 2011년 5월 17일

■ 저자소개 ■



김 태 현
Kim, Tae Hyun

2010년 3월~현재
한양대학교 전자컴퓨터통신공학과
석사과정
2010년 2월 한양대학교 미디어통신공학과
(공학학사)
관심분야 : SDR, LR, WiMAX, 신호처리
E-mail : semoise6@dsplab.hanyang.ac.kr



이 현 석
Lee, Hyun Seok

2011년 9월~현재
삼성전자 무선사업부 SLP Platform
Lab. Application Engineer
2010년 2월 한양대학교 전자통신컴퓨터공학부
(공학석사)
2008년 2월 한양대학교 전자전자컴퓨터공학부
(공학학사)
관심분야 : SDR, 단말 OS Platform
E-mail : hslee.lee@samsung.com



최 승 원
Choi, Seung Won

2002년~현재
HY-SDR 연구센터 센터장
1992년~현재
한양대학교 전자전기공학부 교수
1990년~1992년
일본 우정성 통신연구소 선임연구원
1989년~1990년
ETRI 선임연구원
1988년~1989년
미국 Syracuse대학 전지 및 전산과 교수
1988년 12월 미국 Syracuse대학 전기공학
(공학박사)
1985년 12월 미국 Syracuse대학 컴퓨터공학
(공학석사)
1982년 2월 서울대학교 전자공학 (공학석사)
1980년 2월 한양대학교 전자공학 (공학학사)
관심분야 : SDR, 이동통신, 신호처리
E-mail : choi@dsplab.hanyang.ac.kr