http://dx.doi.org/10.7840/kics.2016.41.9.1141

시변 시스템 추정을 위한 연산량이 적은 가우스 뉴턴 가변 망각인자를 사용하는 RLS 알고리즘

임 준 석◆, 편 용 국°

Low Complexity Gauss Newton Variable Forgetting Factor RLS for Time Varying System Estimation

Jun-Seok Lim◆, Yong-Guk Pyeon°

요 약

일반적으로 RLS 알고리즘에서 비정재성(non-stationary) 환경에서 시간에 따라 변하는 파라메터를 좀 더 잘 추정하기 위해서 가변 망각인자를 사용한다. RLS 알고리즘에서 가변 망각인자를 사용할 때는 연산량이 많이 증가하는 단점이 수반된다. 본 논문에서는 연산량이 적은 가우스 뉴턴 가변망각인자 RLS 알고리즘을 제안한다. 본 방법은 기존 가우스 뉴턴 가변망각인자 RLS와 거의 유사한 성능을 보유하고 있을 뿐만 아니라 부가로 요구되는 연산량을 $O(N^2)$에서 O(N)으로 줄이는 효과도 준다.

Key Words : RLS, Variable forgetting factor, Gauss-Newton

ABSTRACT

In general, a variable forgetting factor is applied to the RLS algorithm for the time-varying parameter estimation in the non-stationary environments. The introduction of a variable forgetting factor to RLS needs heavy additional calculation complexity. We propose a new Gauss Newton variable forgetting factor RLS algorithm which needs small amount of calculation as well as estimates the better parameters in time-varying nonstationary environment. The algorithm performs as good as the conventional Gauss Newton variable forgetting factor RLS and the required additional calculation complexity reduces from $O(N^2)$ to O(N).

Ⅰ. 서 론

RLS (Recursive Least Squares) 알고리즘은 신호처리 및 통신 분야에서 널리 쓰이는 적응 알고리즘 중 하나이다[1-3]. 또한 RLS 알고리즘은 정재성(stationary) 신호 환경에서 좋은 성능과 빠른 적응 속도를 보여주는 알고리즘이지만, 고정된 망각 인자 (혹은 메모리)를 가지고 있기 때문에 비정재성(non-stationary) 신호 환경에서는 좋지 않은 성능을 보여준다.

비정재성 환경에서 계속 변화하는 참 파라메터 값을 추정하는 추적 성능을 높이기 위해서는, RLS 알고리즘의 망각 인자를 가변적으로 조절할 필요가 있다. 이런 가변 망각인자에 대한 기본적인 방법이 Simon Haykin[4]의 책에 소개된 바 있다. 송성욱 등은 가변 망각 인자의 추정 성능을 더욱 향상 시키고자 가우스 뉴턴 형식의 가변 망각인자를 소개한 바 있다[5]. 이는 기존의 가변 망각인자 RLS 알고리즘[4]에 비해 더 좋

◆ First Author : Sejong University Department of Electronic Engineering, jslim@sejong.ac.kr, 종신회원
° Corresponding Author : GangWon State University Department of Information and Communication, pyk12@naver.com, 정회원
논문번호 : KICS2016-06-131, Received June 21, 2016; Revised August 25, 2016; Accepted September 8, 2016

은 성능을 보여주지만, 연산량이 크게 증가하는 단점을 가지고 있다[5].

본 논문에서는 부가 연산량이 적은 가우스 뉴턴 가변 망각인자 RLS 알고리즘을 제안한다. 이를 위해서 기존의 가우스 뉴턴 가변 망각인자 RLS에 대해서 간편화 방법[6][7]을 도입하고, 이를 가우스 뉴턴 형식 알고리즘으로 확장하여 필요한 연산량을 줄인 새로운 알고리즘을 도출한다.

Ⅱ. 가우스 뉴턴 VFF-RLS 방법

송성욱 등[5]이 제안한 이 방법은 기본적으로 재귀적 최소 자승법(Recursive Least Squares: RLS)방법에 기초를 두되, 기존의 가변 망각 인자 RLS 방법과 달리 더욱 빠른 적응을 위해 가우스 뉴턴 형식으로 갱신하는 방법이다. 먼저, 기본적인 RLS 식을 이용하여 가중치 벡터를 계산하는 과정은 다음과 같다[4].

$$e[n]=d[n]-\mathbf{w}^{\mathbf{H}}[n-1]\mathbf{u}[n] \tag{1}$$

$$\mathbf{k}[n]=\frac{\mathbf{P}[n-]\mathbf{u}[n]}{\lambda[n-1]+\mathbf{u}^{\mathbf{H}}[n]\mathbf{P}[n-1]\mathbf{u}[n]} \tag{2}$$

$$\hat{\mathbf{w}}[n]=\hat{\mathbf{w}}[n-1]+\mathbf{k}[n]e^{*}[n] \tag{3}$$

$$\mathbf{P}[n]=\lambda[n-1]^{-1}\times (\mathbf{P}[\mathrm{n}-1]-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}]\mathbf{P}[\mathrm{n}-1]) \tag{4}$$

위의 기본적인 RLS 방법에 가변 망각 인자 $\lambda(n)$를 도입하기 위해서 망각 인자를 MMSE(Minimum mean squared error) 문제를 이용하여 추정할 수 있다. 이때 망각인자 λ에 의한 평균 자승 추정 오류의 기울기는 $-\mathrm{Re}\{\boldsymbol{\psi}^{H}[n-1]\mathbf{u}[n]e^{*}[n]\}$와 같이 나타난다[4]. 여기서 $\boldsymbol{\psi}[n]=\frac{\delta\hat{\boldsymbol{w}}[n]}{\delta\lambda}$ 또 $\mathbf{S}[\mathrm{n}]=\frac{\delta\mathbf{P}[\mathrm{n}]}{\delta\lambda}$를 정의하면, $\boldsymbol{\psi}[n]$을 다음과 같이 재귀적으로 정의할 수 있다 [4,6].

$$\boldsymbol{\psi}[n]=(\mathbf{I}-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}])\boldsymbol{\psi}[\mathrm{n}-1]+\mathbf{S}[\mathrm{n}]\mathbf{u}[\mathrm{n}]e^{*}(\mathrm{n}) \tag{5}$$

$$\begin{aligned}\mathbf{S}[n]=&\lambda[n-1]^{-1}(\mathbf{I}-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}])\\&\times\boldsymbol{S}[n-1](\mathbf{I}-\mathbf{u}[\mathrm{n}]\mathbf{k}^{\mathrm{H}}[\mathrm{n}])\\&+\lambda[n-1]^{-1}\mathbf{k}[n]\mathbf{k}^{\boldsymbol{H}}[n]-\lambda[n-1]^{-1}\mathbf{P}[n].\end{aligned} \tag{6}$$

위의 두 식을 이용하여, 최종적인 가우스 뉴턴 VFF-RLS 방법을 구성하면 다음과 같다[5].

표 1. 가우스 뉴턴형 VFF-RLS[5]
Table 1. Gauss Newton VFF-RLS[5]

$e[n]=d[n]-\mathbf{w}^{\boldsymbol{H}}[n-1]\mathbf{u}[n].$ (7)
$\mathbf{k}[n]=\frac{\mathbf{P}[n-]\mathbf{u}[n]}{\lambda[n-1]+\mathbf{u}^{\boldsymbol{H}}[n]\mathbf{P}[n-1]\mathbf{u}[n]}.$ (8)
$\hat{\mathbf{w}}[n]=\hat{\mathbf{w}}[n-1]+\mathbf{k}[n]e^{*}[n].$ (9)
$\mathbf{P}[n]=\lambda[n-1]^{-1}\times(\mathbf{P}[\mathrm{n}-1]-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}]\mathbf{P}[\mathrm{n}-1]).$ (10)
$\begin{aligned}\mathbf{S}[n]=&\lambda[n-1]^{-1}(\mathbf{I}-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}])\\&\times\boldsymbol{S}[n-1](\mathbf{I}-\mathbf{u}[\mathrm{n}]\mathbf{k}^{\mathrm{H}}[\mathrm{n}])\\&+\lambda[n-1]^{-1}\mathbf{k}[n]\mathbf{k}^{\boldsymbol{H}}[n]-\lambda[n-1]^{-1}\mathbf{P}[n].\end{aligned}$ (11)
$\boldsymbol{\psi}[n]=(\mathbf{I}-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}])\boldsymbol{\psi}[\mathrm{n}-1]+\mathbf{S}[\mathrm{n}]\mathbf{u}[\mathrm{n}]e^{*}(\mathrm{n}).$ (12)
$\begin{aligned}\nabla_{\lambda}'(n)=&(1-\alpha)\nabla_{\lambda}'(n-1)\\&+\alpha\boldsymbol{\psi}^{\boldsymbol{H}}(n-1)\mathbf{u}(n)\mathbf{u}^{H}(n)\boldsymbol{\psi}(n-1).\end{aligned}$ (13)
$\lambda(n)=\lambda(n-1)+\alpha\frac{\mathrm{Re}\{\boldsymbol{\psi}^{H}[n-1]\mathbf{u}[n]e^{*}[n]\}}{\nabla'_{\lambda}(n)}\Big\|_{\lambda_{-}}^{\lambda_{+}}.$ (14)

Ⅲ. 연산량이 적은 가우스 뉴턴 VFF-RLS 방법 (Simplified Gauss Newton VFF-RLS (SGNVFF-RLS))

VFF-RLS 알고리즘의 입력이 i.i.d (independently identically distributed)이고, 그리고 알고리즘의 추정 오류가 통계적으로 독립일 때, 망각인자가 1에 가깝고, 시간 스텝 n이 충분히 크다면 다음과 같이 근사할 수 있다[6,7].

$$(\mathbf{I}-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}])\approx c[\mathrm{n}]\mathbf{I}, \tag{15}$$

여기서, $c[n]=1-\frac{\mathbf{k}^{\mathbf{H}}[n]\mathbf{u}[n]}{L}$. 위의 식을 이용하여 $\mathbf{S}[n]$과 $\boldsymbol{\psi}[n]$, λ를 다시 계산하면 다음과 같다.

$$\overline{\mathbf{S}}[n]=\lambda[n-1]^{-1}\times(|c[n]|^{2}\overline{\mathbf{S}}[n-1]-c[n]\mathbf{P}[\mathrm{n}]). \tag{16}$$

$$\overline{\boldsymbol{\psi}}[n]=c[n]\overline{\boldsymbol{\psi}}[n-1]+\overline{\mathbf{S}}[n]\mathbf{u}[n]e^{*}(n). \tag{17}$$

$$\begin{aligned}\nabla_{\lambda}'(n)=&(1-\alpha)\nabla_{\lambda}'(n-1)\\&+\alpha\overline{\boldsymbol{\psi}^{H}}(n-1)\mathbf{u}(n)\mathbf{u}^{H}(n)\overline{\boldsymbol{\psi}}(n-1).\end{aligned} \tag{18}$$

$$\begin{aligned}\lambda(n)=&\lambda(n-1)\\&+\alpha\frac{\mathrm{Re}\{\overline{\boldsymbol{\psi}^{H}}[n-1]\mathbf{u}[n]e^{*}[n]\}}{\nabla'_{\lambda}(n)}\Big|_{\lambda_{-}}^{\lambda_{+}}\end{aligned} \tag{19}$$

위와 같은 방법으로 연산량을 줄일 수 있지만 좀 더 줄이기 위해서 각 시간 샘플링 간격이 신호의 상관 간격보다 충분이 짧아서 다음 식(20)과 같은 근사적인 관계가 만족한다면,

$$\bar{\mathbf{S}}[n-1]\mathbf{u}[n] \approx \bar{\mathbf{S}}[n-1]\mathbf{u}[n-1], \tag{20}$$

연산량을 좀 더 절약하는 알고리즘을 유도할 수 있다. 즉, $\mathbf{k}[n]=\mathbf{P}[n]\mathbf{u}[n]$[4]인 관계식을 사용하고 $\bar{\mathbf{q}}[n]=\bar{\mathbf{S}}[n]\mathbf{u}[n]$인 관계식을 이용한다면, 다음 식(21)에서 식(23)과 같은 유도 절차를 거쳐서 좀 더 단순화한 알고리즘을 유도할 수 있다.

$$\bar{\mathbf{S}}[n]\mathbf{u}[n] = \lambda[n-1]^{-1}|c[n]|^2 \times \bar{\mathbf{S}}[n-1]\mathbf{u}[n] - \lambda[n-1]^{-1}c[n]\mathbf{P}[\mathrm{n}]\mathbf{u}[\mathrm{n}]. \tag{21}$$

$$\bar{\mathbf{S}}[n]\mathbf{u}[n] \approx \lambda[n-1]^{-1}|c[n]|^2 \times \bar{\mathbf{S}}[n-1]\mathbf{u}[n-1] - \lambda[n-1]^{-1}c[n]\mathbf{P}[\mathrm{n}]\mathbf{u}[\mathrm{n}]. \tag{22}$$

$$\bar{\mathbf{q}}[n] \approx \lambda[n-1]^{-1}|c[n]|^2 \times \bar{\mathbf{q}}[n-1] - \lambda[n-1]^{-1}c[n]\mathbf{k}[\mathrm{n}]. \tag{23}$$

위와 같은 단순화 과정을 거친 새로운 알고리즘을 표 2에 정리하였다.

표 2. 간단형 가우스 뉴턴형 VFF-RLS
Table 2. Simplified Gauss Newton VFF-RLS

$e[n]=d[n]-\mathbf{w}^{H}[n-1]\mathbf{u}[n].$ (24)
$\mathbf{k}[n]=\dfrac{\mathbf{P}[n-]\mathbf{u}[n]}{\lambda[n-1]+\mathbf{u}^{\mathrm{H}}[n]\mathbf{P}[n-1]\mathbf{u}[n]}.$ (25)
$\hat{\mathbf{w}}[n]=\hat{\mathbf{w}}[n-1]+\mathbf{k}[n]e^{*}[n]$ (26)
$\mathbf{P}[n]=\lambda[n-1]^{-1} \times(\mathbf{P}[\mathrm{n}-1]-\mathbf{k}[\mathrm{n}]\mathbf{u}^{\mathrm{H}}[\mathrm{n}]\mathbf{P}[\mathrm{n}-1]).$ (27)
$c[n]=1-\dfrac{\mathbf{k}^{\mathbf{H}}[n]\mathbf{u}[n]}{L}.$ (28)
$\bar{\mathbf{q}}[n] \approx \lambda[n-1]^{-1}\lvert c[n]\rvert^2 \times \bar{\mathbf{q}}[n-1]-\lambda[n-1]^{-1}c[n]\mathbf{k}[\mathrm{n}].$ (29)
$\bar{\boldsymbol{\psi}}[n]=c[n]\bar{\boldsymbol{\psi}}[n-1]+\bar{\mathbf{q}}[n]e^{*}(n).$ (30)
$\nabla_{\lambda}{}'(n)=(1-\alpha)\nabla_{\lambda}{}'(n-1) +\alpha\overline{\boldsymbol{\psi}^{H}}(n-1)\mathbf{u}(n)\mathbf{u}^{H}(n)\bar{\boldsymbol{\psi}}(n-1).$ (31)
$\lambda(n)=\lambda(n-1) +\alpha\dfrac{\mathrm{Re}\{\overline{\boldsymbol{\psi}^{H}}[n-1]\mathbf{u}[n]e^{*}[n]\}}{\nabla'_{\lambda}(n)}\Big\vert_{\lambda_{-}}^{\lambda_{+}}$ (33)

표 3. 계산 복잡도 비교표
Table 3. Complexity comparison

	Total Complex Multiplication	Additionally Required Multiplication for Forgetting Factor update
RLS[4]	$2.5N^2+3N$	0
VFF-RLS[4]	$9N^2+7N$	$6.5N^2+4N$
GNVFF-RLS[5]	$10N^2+7N$	$7.5N^2+4N$
SGNVFF-RLS	$2.5N^2+7N$	4N

단순화 방법을 통해서 얻는 새로운 알고리즘과 기존의 알고리즘의 복잡도를 비교하기 위해서 각 알고리즘에서 필요로 하는 곱셈의 횟수를 척도로 하여 표3에 비교하였다.

표 3을 보면 제안된 단순화 알고리즘을 이용하면 고정형 망각인자를 사용하는 전통적인 RLS알고리즘에 대해서 4N 만큼의 곱셈 복잡도만을 더 요구하는 것을 알 수 있다. 이는 기존에 알려진 가변 망각인자 RLS가 $6.5N^2+4N$의 곱셈 복잡도를 더 요구하는 것에 비해서 매우 큰 차로 복잡도를 단순화시킴을 알 수 있다.

Ⅳ. 실험 결과

제안된 알고리즘의 성능 보기 위해서 두 개의 시뮬레이션 실험을 하였다. 첫 번째 시뮬레이션 실험은 두 개의 서로 다른 채널이 특정 시간에 갑자기 바뀌는 경우에 채널 추정 성능을 보이는 것이고, 두 번째 시뮬레이션 실험은 일정한 시변성을 갖는 채널에 대해서 여러 신호대 잡음비에 대한 채널 추정 성능을 보인다.

4.1. 갑자기 변하는 채널 추정

제안한 방법이 GNVFF-RLS[5]과 유사한 성능을 낸다는 것을 보이기 위한 일환으로 다음과 같은 두 개의 채널을 상정하고 B1 채널에서 시작하여 300 샘플 스텝에서 B2 채널로 바뀌는 경우에 채널 추정 성능을 비교하여 보인다.

$$\mathrm{B1} = [1, 0.8668, -0.4764, 0.2070]. \tag{34}$$

$$\mathrm{B2} = [1, -0.8326, 0.6656, -0.7153]. \tag{35}$$

또 이 시뮬레이션에서 신호대 잡음비는 25dB로 설정하여 500회 반복 실험을 하였다. 다음 그림 1에 비

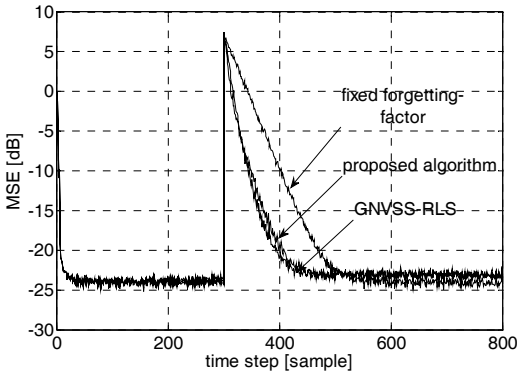

그림 1. 갑자기 변하는 채널 추정 성능 비교
Fig. 1. Channel estimation performance in the abruptly changing channel environments

교 실험 결과한 결과를 보였다. 그림에서 추정의 성능을 MSE (mean square error)를 dB 단위로 보였다.

그림 1에서 300 샘플 스텝이후 채널 추정 성능을 보면 비교 대상인 GNVFF-RLS[5]과 제안된 SGNVFF-RLS 알고리즘이 거의 비슷한 성능을 보임을 알 수 있다.

4.2. 여러 신호대 잡음비에서 연속적으로 변하는 채널 추정

여러 신호대 잡음비에서 연속적으로 변하는 채널 추정을 통해서 제안된 알고리즘이 GNVFF-RLS[5]과 유사한 성능을 보인다는 것을 확인하기 위해서 GNVFF-RLS이 제안된 논문[5]과 같이 2차 변수 중 한 변수가 시변인 환경 하에서 시뮬레이션을 실시하였다.

$$d(n)=\mathbf{h}^T(n)\mathbf{u}(n)+v(n), \tag{36}$$

여기서 $\mathbf{h}^T(n)=[a(n)\,1]$, $a(n)=1.6\cos(2\pi f_m n)$, $\mathbf{u}^T(n)=[s(n-1)\;\; s(n-2)]$, $s(0)=1$. 또 추정 성능을 보이기 위한 척도로 다음과 같이 정의되는 MSD (mean standard deviation)을 사용한다.

$$\text{MSD}=E[\mathbf{h}(n)-\hat{\mathbf{h}}(n)], \tag{37}$$

여기서 $\hat{\mathbf{h}}(n)$는 추정된 채널 응답이다. 그리고 f_m=0.005로 설정하였다. 이 환경에서 얻은 결과는 그림 2에 정리하여 그렸다. 그리고 표 4에는 그림 2의 결과를 정리하기 위해서 각 신호대 잡음마다의 각 알고리즘마다의 추정 성능 차이를 수치로 정리하였다.

그림 2와 표 4를 보면 제안된 방법이 비교대상인 GNVFF-RLS[5]과 신호대 잡음비 5dB부터 25dB 사이

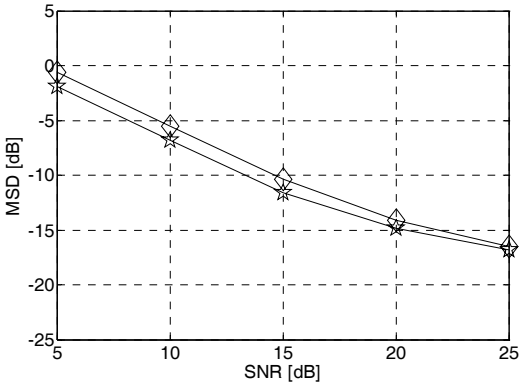

그림 2. f_m=0.05일 때 여러 신호대 잡음비에서 추정 정확도 비교
Fig. 2. Estimation accuracy comparison with MSD in various SNR at f_m=0.005
(-◇-: the proposed algorithm, -☆-: GNVFF-RLS algorithm)

표 4. GNVFF-RLS와 제안된 알고리즘 사이의 추정 정확도 차이
Table 4. Estimation accuracy difference between GNVFF-RLS and the proposed algorithm

SNR [dB]	5	10	15	20	25
Estimation accuracy difference between GNVFF-RLS and the proposed algorithm [dB]	-1.3	-1.3	-1.2	-0.7	-0.3

에서 약 1 dB 이내에서 성능을 유사하게 유지됨을 보이고 있다.

V. 결 론

본 논문에서는 비정재성 환경에서 추정 성능이 우수한 알고리즘을 제안하였다. 제안된 알고리즘은 비정재성 환경에서 추정 성능이 우수하다고 알려진 GNVFF-RLS과 유사한 성능을 유지하면서 연산량은 $O(N^2)$에서 O(N)으로 줄어들도록 만들었다. 시뮬레이션을 통해서 제안된 알고리즘이 비교대상인 GNVFF-RLS과 유사한 추정 성능을 보임을 확인하였다.

References

[1] S. Han, C. Song, and J. Choi, "Interference cancellation for wireless LAN systems using full duplex communications," *J. KICS*, vol. 40, no. 12, pp. 2353-2362, Dec. 2015.

[2] J. Lim and Y. Pyeon, “Kernel RLS algorithm using variable forgetting factor,” *J. KICS*, vol. 40, no. 9, pp. 1793-1801, Sept. 2105.

[3] M. Choi and S. Lee, “Comparison study of channel estimation algorithm for 4S maritime communications,” *J. KICS*, vol. 38, no. 3, pp. 288-293, Mar. 2013.

[4] Simon Haykin, *Adaptive filter theory*, Prentice-Hall, Inc., 4th Ed., 2002.

[5] S. Song, J. Lim, S. Baek, and K. Sung, “Gauss newton variable forgetting factor recursive least squares for time varying parameter tracking,” *Electronics Lett.,* vol. 36, no. 11, pp. 988-990, Nov. 2000.

[6] S. Song, “Self-tuning adaptive algorithm and applications,” Ph.D. dissertation, Seoul National Univ. 2003.

[7] S. Song and K. Sung, “Reduced complexity self-tuning adaptive algorithms in application to channel estimation,” *IEEE Trans. Commun.,* vol. 55, no. 8, pp. 1448-1452, Aug. 2007.

임 준 석 (Jun-Seok Lim)

1986년 2월 : 서울대학교 전자공학과 학사 졸업

1988년 2월 : 서울대학교 전자공학과 석사 졸업

1996년 8월 : 서울대학교 전자공학과 박사 졸업

1996년 7월~1997년 10월 : LG 종합기술원

1998년 3월~현재 : 세종대학교 전자정보통신공학과 교수

<관심분야> 신호처리

편 용 국 (Yong-Guk Pyeon)

1993년 2월 : 강원대학교 전자공학과 학사 졸업

1996년 2월 : 관동대학교 전자공학과 석사 졸업

2004년 2월 : 세종대학교 전자공학과 박사 졸업

2004년 9월~현재 : 강원도립대학교 정보통신과 조교수

<관심분야> 신호처리