

# 표준 이더넷 기반의 군사용 지상무인차량 실시간 네트워크 구현

김도종\*<sup>1</sup>, 김남곤\*<sup>2</sup>, 양동원\*<sup>3</sup>

## Real Time Network Implementation for Military Unmanned Ground Vehicle based on Standard Ethernet

Dojong Kim\*<sup>1</sup>, Namgon Kim\*<sup>2</sup>, and DongWon Yang\*<sup>3</sup>

### 요 약

이더넷 기술은 높은 대역폭과 신뢰성을 제공하여 산업용과 가정용으로 널리 활용되고 있으며, 대용량 데이터 전송에 적합한 유망한 후보로 주목을 받고 있으나 전송지연 등의 불확실한 전송특성으로 인하여 실시간 처리에 적절치 않은 것으로 인식되어 왔다. 본 논문에서는 표준이더넷을 군사용 지상무인시스템에 적용하여 대용량신호 전송과 실시간 전송 구현이 가능함을 소개한다. CPU 코어를 쉴딩하고 네트워크처리 전용으로 활용하면 전송지연을 줄이고 안정적인 전송 성능을 얻을 수 있음을 보였고, 정적라우팅, 스위치적용, QoS 등의 기법들과 같이 실제 무인차량에 적용하여 전송불확실성 최소화화 차량네트워크의 실시간성이 보장될 수 있음을 입증하였다.

### Abstract

Ethernet technology has shown high reliability and bandwidth performance for commercial and industrial systems. Although it is regarded as a promising candidate for massive data transmission, it is known to be unsuitable for a realtime system due to an uncertainty of transmission latency. In this paper we have applied standard Ethernet to military unmanned ground vehicle(UGV) and shown that the network could overcome the problems of massive data transmission and realtime operation. By adopting CPU core shielding and various affinity options, we can reduce the transmission latency and make our system more stable. Above technique with static routing, full duplex switch, QoS is implemented into UGV system and real time operation of the vehicle is proved.

### Keywords

standard ethernet, network, unmanned ground vehicle, shielding, affinity, latency, jittering

---

\* 국방과학연구소 지상기술연구원(\*<sup>1</sup> 교신저자)  
- ORCID<sup>1</sup>: <http://orcid.org/0000-0002-4886-1403>  
- ORCID<sup>2</sup>: <http://orcid.org/0000-0002-1038-3768>  
- ORCID<sup>3</sup>: <http://orcid.org/0000-0002-6886-2008>

.. Received: Jul. 14, 2021, Revised: Sep. 30, 2021, Accepted: Oct. 03, 2021  
.. Corresponding Author: Dojong Kim  
Unmanned Ground System PMO Team 1, Ground Technology Research  
Institute, Yuseong P.O.Box 35, Daejeon, 34186, Korea  
Tel.: +82-42-821-0704, Email: [djkim@add.re.kr](mailto:djkim@add.re.kr)

## 1. 서 론

기존 군용차량에서는 1553, RS422, CAN(Control Area Network), LVDS(Low Voltage Differential Signaling) 등의 표준버스를 활용하여 차량 네트워크를 구현하였으나 새로운 무기체계 적용에는 한계를 나타내고 있다. 최근에 원격 또는 자율로 운용되는 무인로봇 체계의 등장으로 차량내부의 컴퓨터들이 증가하였고, 자율기능 추가, 안전/보안 기능 강화 및 영상/음성/레이다/라이다 등의 대용량 신호처리 요구증가와 확장성, 상호운용성 등에 대한 한계가 발생함에 따라 군사용 차량네트워크에 대한 새로운 패러다임이 요구되고 있다[1].

IEEE 표준이며 민수/산업영역에서 활용성이 높고 자동차분야로 확산이 예상되는 이더넷 기술은 SW/부품의 재사용성이 우수하고 높은 대역폭 성능을 제공하여 상기 문제를 극복할 수 있는 유망한 후보로 주목을 받고 있다. 이더넷은 경량 디바이스, 저가, 신뢰성, 높은 대역폭 을 가지며 유선/무선 네트워크 접속도 용이하게 할 수 있다는 특성을 가진다. 또한, Flexible 기술적용으로 빠른 플랫폼 변경 및 확장이 용이하고, TCP/IP 등 상위 레이어 프로토콜을 포함한 기 개발 모듈의 활용으로 개발절차도 단순하여 가정용, 산업용, 차량/항공/군사 용 등 미래의 다양한 요구조건 충족이 가능한 기술이다.

IEEE 표준 이더넷에는 랜덤한 전송지연, 잠재적인 전송에러 등의 현안을 내재하고 있다. 전송용량은 크지만 불확실 전송성능에 의해 실시간 응용체계에 부적합하여 실시간 네트워킹 분야에서 배제되어 왔으나, 최근 컴퓨팅능력 증가, 전송능력 향상, 확장성 용이 및 가격저하 등에 힘입어 실시간 네트워킹 시스템 적용에 대한 노력이 진행 중이다[2][3].

본 논문에서는 군사용 지상무인차량에 표준이더넷을 적용하여 실시간 네트워크 성능을 구현한 방법과 결과를 소개한다. 차량네트워크는 확장성, 정비성, 개발용이성, 표준화 등을 고려하여 이더넷을 백본으로 구성하였다. OS 기반의 상위레이어 프로토콜 적용으로 개발절차를 단순화 하였고, 실시간 OS(RedHawk) 적용{RedHawk OS는 RedHat Linux를 기반으로 Kernel을 수정(Modified linux kernel) 탑재 및 실시간처리기능을 강화하여 개발된 RTOS}, 전

이중(Full-Duplex) 방식 스위치 적용, TCP/UDP 프로토콜의 선택활용, 스위치 버퍼조정, 정적라우팅(Static routing) 등의 적용으로 표준 이더넷의 장점을 살리면서 데이터 전송의 불확실성 최소화와 성능향상을 구현하였다. 멀티코어 CPU를 기반으로 특정 코어를 네트워크처리 전용으로 설정하여 안정적이며 전송지연 성능을 향상시킬 수 있었다.

전송성능 검증을 위해 단계적으로 시험을 수행하였다. 먼저, 단위네트워크 시험에서는 CPU 코어 전용설정과 최적화기능을 통하여 안정적이며 지연성능향상이 가능함을 보였고, 실제 무인차량에 탑재 및 N:N으로 동작하는 통합네트워크 상태에서 성능확인을 통하여 이더넷으로 지상무인차량의 실시간네트워크구현이 가능함을 보여준다.

본 논문의 기여도는, 표준이더넷의 단점으로 알려진 불확실한 전송성능과 실시간 시스템에 적용문제들에 대한 극복방안을 소개하였고, 단위네트워크 성능시험과 실제 지상무인차량 시스템에 적용하여 실시간 동작 구현이 가능함을 보였다. 논문의 구성은 다음과 같다. 2장에서는 이더넷 활용과 실시간처리 현안을 소개하고, 3장에서는 지상무인차량의 네트워크 구성, 성능요구조건 분석 및 CPU 코어설정, 최적화기능을 활용한 단위네트워크 성능시험 결과를 소개한다. 4장에서는 지상무인차량의 통합네트워크에 대한 시험결과를 소개하고 마지막으로 5장에서는 결론 및 네트워크 활용의 고려사항에 대하여 기술한다.

## II. 이더넷 활용과 실시간 전송

### 2.1 활용분야

이더넷의 시초는 1972년 건물내 서버, 스테이션, 주변장치 연결을 위한 공통버스(Common bus) 개발을 목적으로 시작되었다. 이후, CSMA/CD 기술이 접목되어 발전이 가속되었고 1985년에 IEEE에서 이더넷이라는 이름 대신에 802.3 CSMA/CD 라는 기술로 표준화를 시작하여 본격적으로 확산이 시작되었다. 적용분야도 빌딩내부의 광역정보 시스템구축을 시작으로 차량/해상/항공/산업 등의 임베디드 네트워크분야 및 멀티미디어 네트워크, 프라이빗

(Private), 가정용 네트워크분야 등으로 확대 적용이 되고 있다. 이더넷 성능의 발전추세는 그림 1과 같다[4]. 산업용 네트워크는 다양한 종류의 전용 Fieldbus를 사용해오고 있었으나, 네트워크 접속장치 수의 제한에 따른 확장성 문제와 다양한 기술적 솔루션들과의 상호운용성이 보장되지 않는 점 등으로 공장 자동화를 준비하는 분야에서는 이더넷으로 대체하는 것에 관심을 두고 있다. TCP/IP 기반의 이더넷이 네트워크 인프라구축 비용을 절감하고 어느 곳에서든 어떤 장비 하고도 접속이 가능한 특징에 힘을 입어 점차적으로 대체 변화를 시도하고 있다.

산업용 네트워크는 기본적으로 센서, 액츄에이터 및 제어기와 정보를 주고받아야 하며, 공장내부의 많은 장치들이 상호 연결되어야 하고 사무실내에서도 모니터링/제어가 가능해야한다. 따라서, 산업용 네트워크의 성능은 제어장치 수량, 데이터속도, 네트워크 크기, 응답시간, 부하크기 및 데이터교환 주기 등에 의해 결정된다. 일반용 이더넷에 비해 산업용 이더넷이 가져야 하는 대표적인 특징은 시간동기화, 실시간 동작, 높은 가용성(High availability) 과 전기잡음, 진동, 온도, 충격 등에 대한 내구성이다 [5][6].

최근에 차량제어 SW가 급격히 증가됨에 따라 차량네트워크도 중요한 현안이 되고 있다. 일반적인 ECU(차량전자장치)의 구성은 센서, 액츄에이터 및 마이크로 콘트롤러 등으로 구성되며 차량제어 (Powertrain, Chassis, Body, Comfort), 인포테인(DVD Player, Navigation, 원격진단), 통신 등의 영역으로 구분할 수 있다. 최근의 차량에는 70여개 이상의 ECU를 탑재하고, ECU 간에는 네트워크를 통하여 데이터를 송수신하고 있다.

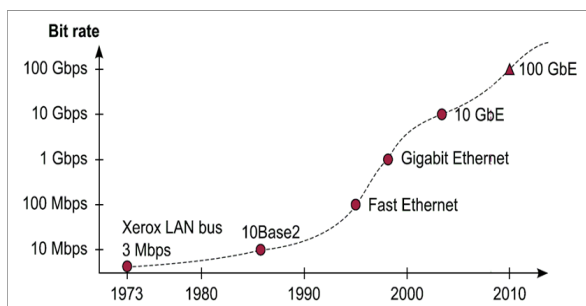


그림 1. 이더넷 발전  
Fig. 1. Ethernet evolution

기존에 사용 중인 네트워크는 LIN, CAN, MOST (Media Oriented System Transport) 등으로 복잡성, 고비용, 확장성, 정비성 및 전송용량 제한 등에 따라 문제에 직면하고 있다.

전송능력 향상이 추가적비용 부담으로 작용함에 따라 생산/공급자들은 차량요구 조건을 충족하는 저가의 네트워크 기술분야로 눈을 돌리고 있으며, 개발시간 단축에도 중점을 두어 차량전용 기술개발 보다 표준화되어 대량으로 유통되는 기술을 찾는 중이고 이미, 이더넷을 적절한 대안으로 판단하여 준비 중에 있다[7][8]. 차량용 네트워크 시스템에 대한 주요 요구사항은 전송속도, 안전, 신뢰성, 설치/사용 용이성, 온도, 전자파, 크기/중량 등이다.

군사용 차량네트워크는 1553, RS422, CAN, LVDS 등의 표준버스를 기반으로 활용하였다. 최근에 지상무인로봇 체계의 등장으로 차량내부의 컴퓨터 증가, 원격/자율주행, 안전/보안 기능 강화 및 영상/음성/레이다/라이다 등의 대용량 신호처리 요구 증가와 확장성, 상호운용성 등에 대한 한계가 발생함에 따라 군사용 차량네트워크에 대해서도 새로운 패러다임이 요구되고 있다.

군사용 네트워크는 실시간성, 안전, 보안, 확장성 (Scalability), 정비성(Maintainability), 전자기파, 환경 조건 등이 요구되며, 개발용이성, 비용도 중요한 요소가 되며 COTS(Commercial Off the Shelf) 기술을 적용한 비용절감, 표준화기술 사용 및 활용소스 (Vendor) 다변화 등이 주요한 관심사항이 되고 있고, 이에 따라 이더넷이 주요대상으로 고려되고 있다.

## 2.2 실시간 네트워킹 현안

자동화를 위한 산업용, 차량제어용 및 군사용 네트워크의 공통요구사항 중의 하나가 실시간성이다. 실시간 처리는 단순히 연산이 빠르다는 의미보다 시간과 동기화 데이터를 처리하되 주어진 시간제한을 초과하지 않게 동작할 수 있음을 말하며 아래의 조건을 만족하면 실시간(Real-time)이라고 한다.

$$P(A|B) = 1, \tag{1}$$

$$A = (rx + \delta p + tx) < B, B = T2 - T1$$

여기서, A는 응답시간, B는 주기를 나타내며 세부적인 구성은 그림 2와 같다. 그림 2에서 rx는 입력 명령 또는 센서신호의 수신지연시간,  $\delta p$ 는 Task 처리시간, tx는 처리결과 송신지연시간 이다. 네트워크 관점의 실시간 처리는 규정된 시간내 처리대상 데이터의 입출력 전송처리가 가능하면 실시간 네트워크로 볼 수 있다.

표준 이더넷을 실시간 네트워킹으로 활용에 가장 큰 문제점은 CSMA/CD 방식의 적용에 따른 데이터 충돌과 전송지연이다. 전이중 방식의 스위치를 적용하면 데이터 충돌문제 극복은 가능하지만 데이터를 스위치 내부 버퍼에 저장 및 대기(Queue) 과정을 거치고, 목적지 전송을 위해 하나의 스위치를 거치는 경우에는 버퍼 오버플로우(Buffer overflow)등의 발생으로 새로운 전송지연 또는 패킷손실이 발생할 수 있다. 이더넷의 프로토콜(TCP/UDP/IP) 처리지연, 네트워크카드(NIC)의 지연/지터링과 비정적 라우팅(Non-static routing) 방식 등에 의한 패킷전송 시간의 예측 불확실성은 실시간 네트워크로 활용을 어렵게 하는 요인으로 작용한다.

이러한 현안극복 및 실시간성 확보를 위해 그림 3과 같이 다양한 네트워크 구조들이 제안되었으며, 대표적인 네트워크는 Modbus-TCP, Ethernet Powerlink, EtherCAT, TCnet, TTEthernet, CC-Link IE Field, Profinet, EtherNet/IP 등이 있다[2][4].

그림 3의 Category A는 표준 이더넷규약과 스위치를 적용하여 Soft Real time(100ms 지연수준)을 목표로 하는 방식이며, B는 Hard Real time(1~10ms 지연)을 목표로 전용 데이터 처리 프로토콜을 추가하는 방식이다. C는 0.24ms~1ms 지연 및 1us 이하의 지연을 목표로 하여 전용 프로토콜과 전용 네트워크카드를 구현하는 방식이다[2].

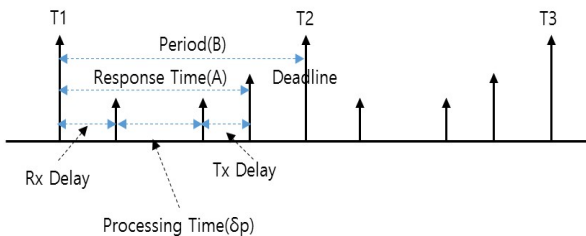


그림 2. 실시간 동작  
Fig. 2. Real time operation

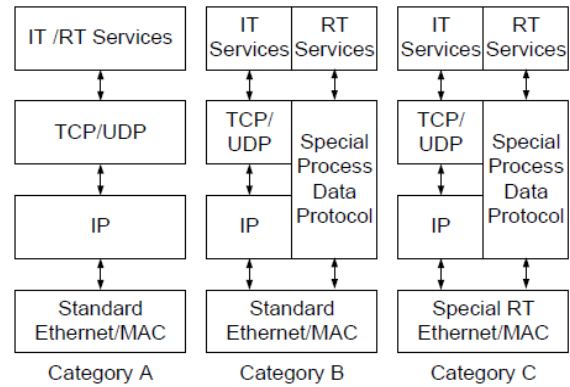


그림 3. 실시간 네트워킹 구현방식  
Fig. 3. Real time implementation approach

본 논문에서는 Category A 방식을 적용하여 표준 네트워크 장점을 최대한 활용하면서 실시간 동작이 보장되도록 구현할 수 있는 방법 및 결과에 대해 소개한다.

### III. 무인차량 구성 및 단위네트워크 시험

#### 3.1 무인차량 네트워크구성

본 논문에서 소개하는 무인차량시스템은 한 대의 통제차량과 2대의 무인차량으로 구성된다. 통제차량과 무인차량은 무선으로 연동되며, 각각의 차량내부는 다수의 노드로 구성되고 이들은 기가비트 이더넷을 활용한 유선네트워크로 구성된다. 유선네트워크는 데이터망과 기록망을 분리 구현하여 무인차량 외부에서 접속은 기록망을 활용하고, 무인차량 내부 제어정보는 데이터망을 활용할 수 있게 하여 보안과 시스템성능 확보를 고려하였으며 그림 4와 같다. 이때, 외부접속은 무인차량과 통제차량에 직접연결하여 차량내부 전자장치에 대한 SW Update, Debugging, Data Logging 등을 수행하며, 데이터망과 분리 운용하여 보안성 유지를 목적으로 한다.

#### 3.2 네트워크 요구성능

일반적인 차량용 네트워크의 성능은 전송용량, 실시간성, 신뢰성, 확장성, self-configuration 등을 포함하며 응용체계에 따라 성능에 대한 요구수준이 달라진다.

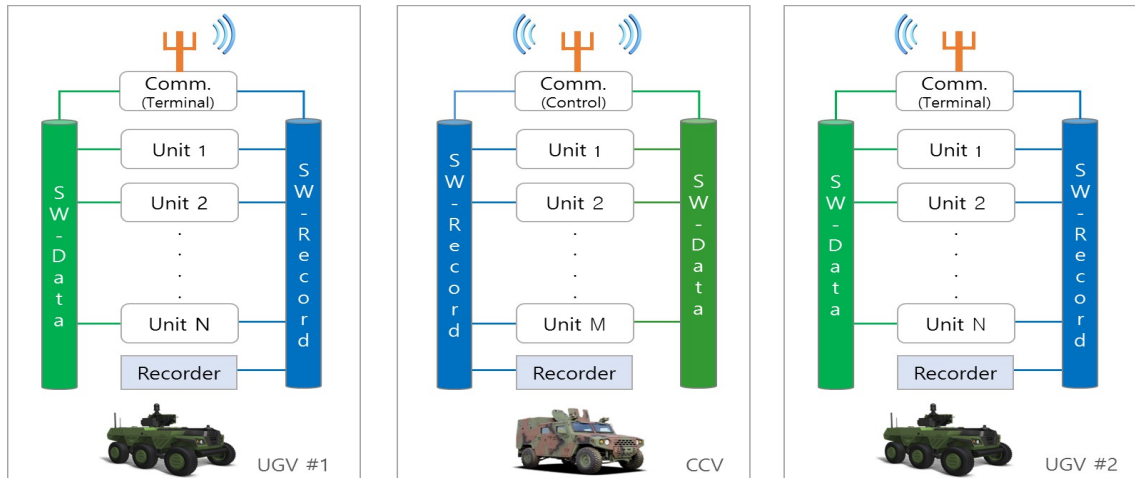


그림 4. 무인차량 네트워크  
Fig. 4. Network of unmanned ground vehicle

본 연구에서 소개하는 지상무인차량의 네트워크는 영상, 센서(라이다/레이다 등), 제어, 상태 등의 신호들을 공통네트워크를 통해 송수신이 가능해야 한다. 대용량전송 및 실시간성을 동시에 해소할 수 있어야하며, 유효 전송성능이 500Mbps 이상이어야 한다.

차량내부의 주요장치(감시/무장/자율주행장치 등)에 대해서는 200Hz, 100Hz, 50Hz, 10Hz의 실시간 제어를 처리할 수 있어야하며, 단위장치 간 패킷의 최대 전송지연시간은 0.5ms 이하(200Hz 실시간처리를 고려하여 프레임시간 5ms의 10% 이내로 설정), 전송누락은 0.01% 이하를 목표로 한다. 또한, 향후 무인차량 수량 및 차량내부 장치들의 수량증가 등에 대해 유연한 확장성이 제공되어야 한다.

### 3.3 단위네트워크(1:1) 성능시험

성능시험은 먼저, 단위네트워크 시험을 통하여 최상으로 구현 가능한 전송성능을 확인하고, 실제 무인차량에 탑재하여 시스템상태에서의 성능을 평가한다. 단위네트워크 시험은 그림 5와 같이 On-board 이더넷이 장착된 SBC 보드 2장과 스위치로 구성하여 수행하였다. 각각의 SBC 보드에는 멀티코어 CPU가 장착되었으며 이더넷 컨트롤러, 송수신 SW모듈, RedHawk OS를 탑재하였고, GLAN 스위치를 통해 연결하였다.

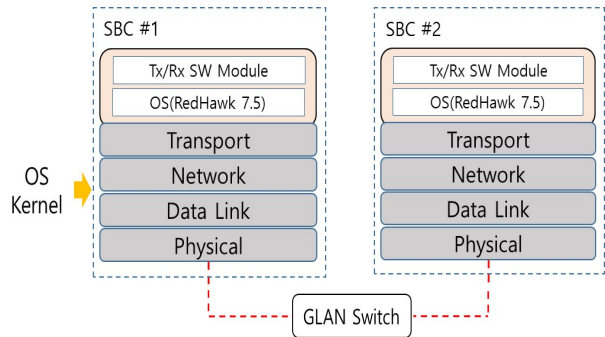


그림 5. 1:1 성능시험  
Fig. 5. 1:1 Performance test

전송성능은 전송지연시간(평균/최대/최소)과 지터링(표준편차)을 기준으로 평가하였다. 네트워크 성능평가를 위한 상용SW모듈이(Pingpong, Netperf, Sfint) 다양하게 제공되고 있으나, 본 논문에서는 상용SW와 성능은 동일하게 유지하며 성능평가에 적합토록 전송주기/데이터크기/개수 등의 제어가 가능한 자체 개발 송수신 SW 모듈을 탑재 및 활용하였다.

성능시험은 특정 CPU 코어를 네트워크처리 전용으로 할당하여 네트워크 인터럽트 또는 송수신 SW만 수행될 수 있도록 하고, 네트워크 최적화를 위한 network-latency 튜닝 프로파일(Tuning profile) 기능 적용 유무에 따른 성능을 측정하였다.

또한, GLAN 스위치 적용유무, 전송프로토콜(UDP/TCP) 종류, 데이터크기(64~ 1518Byte), 전송주기 등 다양한 조건별 시험을 수행하였으며 시험환경 구성은 표 1과 같다.

실험에 적용된 UDP 데이터 크기 변경은 User Data Size를 조정하여 구현하였다.

본 논문에서는 CPU 3번 코어에 네트워크처리 인터럽트 할당(Int-Affinity), 송수신처리 SW 할당(Proc-Affinity), network-latency 튜닝 프로파일의 적용 유무 조건에서 전송주기별 시험을 수행하였다. 세부적인 시험조건은 표 2와 같으며, 표내에서 ‘1’은 기능적용, ‘0’은 기능 미적용을 표시한다.

표 1. 시험환경  
Table 1. Test environment

Type	Functions	
SBC	CPU	Intel Xeon E3-1505L, 2GHz 4 Core, L1/L2/L3 Cache
	Memory	16Gb
	NIC	Intel I210 gigabit controller
OS	RedHawk	ReHawk7.5(Kernel:4.9.98-rt76)
GLAN switch	ConnectTech Xtreme/GbE 24-port switch	
Data size(Byte)	64, 256, 512, 768, 1024, 1518	

표 2. 적용옵션 및 시험조건  
Table 2. Option and test condition

Type	Option	Test condition							
1KHz	Int-Affinity	0	0	0	0	1	1	1	1
	Proc-Affinity	0	0	1	1	0	0	1	1
	Tuning profile	0	1	0	1	0	1	0	1
200Hz	Int-Affinity	0	0	0	0	1	1	1	1
	Proc-Affinity	0	0	1	1	0	0	1	1
	Tuning profile	0	1	0	1	0	1	0	1

CPU 3번 코어를 사용하기 위해서 해당코어에 모든 인터럽트, 프로세스 할당금지처리(Core shielding)를 해야 한다. 이후 전용처리가 필요한 인터럽트, 프로세스를 할당 하고 또한, 네트워크 전송 지연 최적화를 위해 network-latency 프로파일을 적용하며 (튜닝 프로파일) 이를 위한 명령어는 표 3과 같다[9].

성능시험은 10,000개의 UDP 데이터를 크기별로 주기전송(1KHz, 200Hz) 하고 성능을 측정하였다. 표 2의 8가지 조건으로 시험을 수행하였으며 시험결과는 5가지 조건(옵션 미적용 1, 단일옵션 적용 3, 통합옵션 적용 1)에 대해서 그림 6~9에 나타내었다. 나머지 조건(단일옵션의 조합 3종)에 대한 성능은 단일옵션 적용 시 보다는 우수하나 통합옵션적용 경우보다는 성능이 저하되는 현상을 확인하였으며 성능분석 결과는 별도로 수록하지 않았다.

그림 6~9는 데이터 크기별(64/256/512/768/1024/1518) 1KHz 주기 전송한 결과에 대해 전송시간(RTT/2)의 평균/최대/최소/편차를 나타내었다.

표 3. 옵션설정 명령어  
Table 3. Option command

Option	Command
Core shielding	Shield -I3 -p3 -a3 (CPU Core#3)
Int-Affinity	echo \${cpu} > /proc/irq/\${irq}/smp_affinity
Proc-Affinity	void CPU_SET(int cpu, cpu_set_t *set);
Tuning profile	tuned-adm profile network-latency

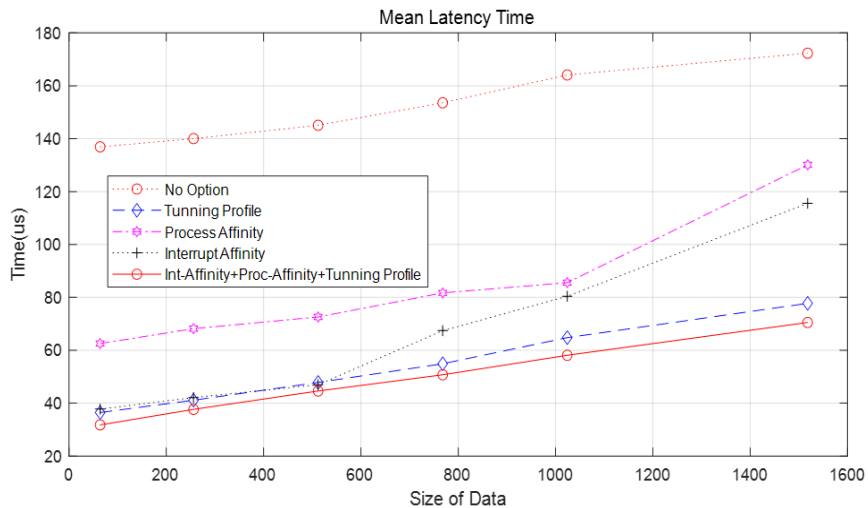


그림 6. 평균지연  
Fig. 6. Mean latency

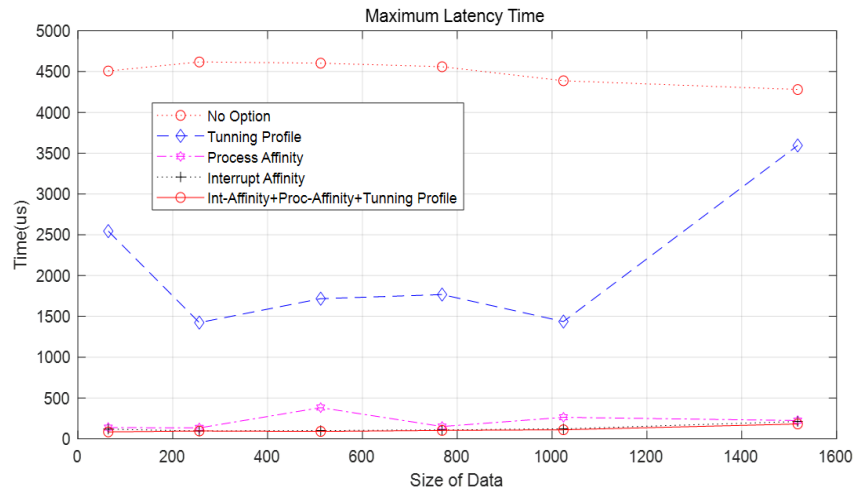


그림 7. 최대지연  
Fig. 7. Maximum latency

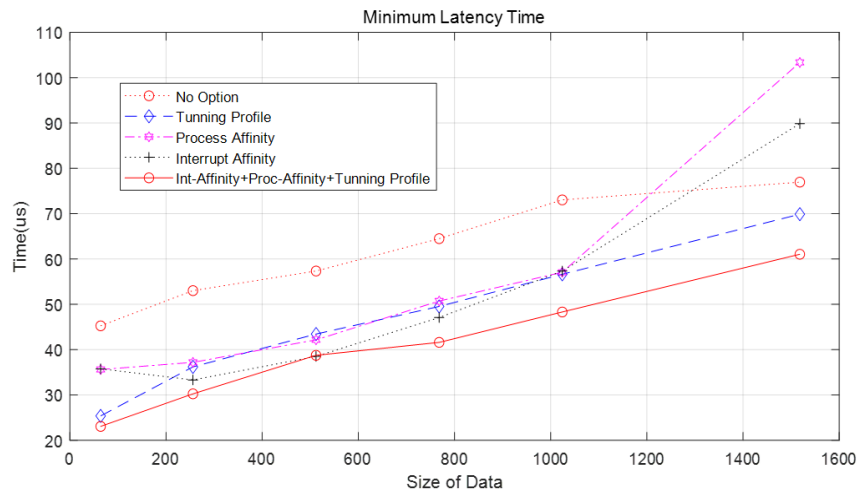


그림 8. 최소지연  
Fig. 8. Minimum latency

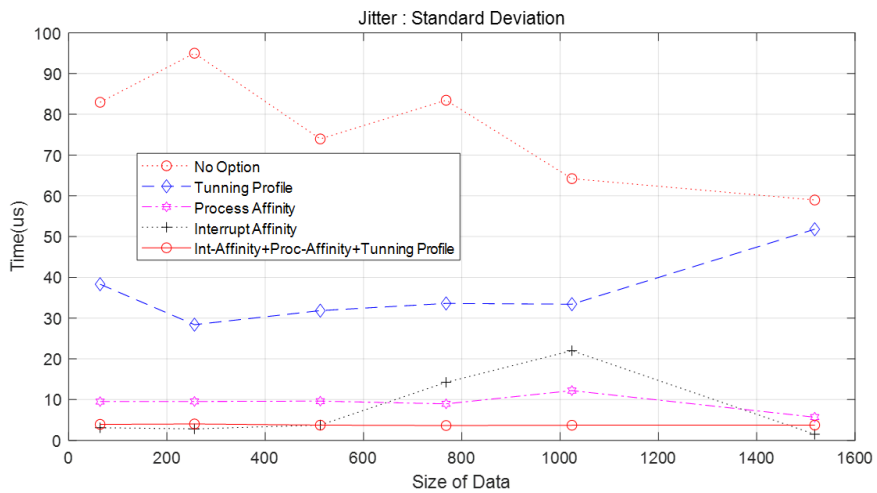


그림 9. 지터링(지연편차)  
Fig. 9. Jittering(standard deviation)

옵션을 적용하지 않을 경우 평균지연이 140~170 $\mu$ s, 최대지연이 4000~ 5000 $\mu$ s, 편차가 60~100 $\mu$ s 수준으로 큰 값을 나타낸다. 전송 지터링값이 크고, 최대지연이 4000 $\mu$ s 이상으로 200Hz 처리를 위한 실시간성 보장은 거의 불가능함을 알 수 있다. 네트워크성능 최적화를 위한 network-latency 튜닝 프로파일 기능만 적용할 경우 평균전송 시간은 30~80 $\mu$ s 수준으로 우수함이 확인되나 최대지연시간은 1500~3500 $\mu$ s 수준으로 여전히 실시간 제어에 적절하지 않음을 알 수 있다. 이는 데이터 송수신 동안 빈도가 높지는 않으나 간헐적으로 과도한 지연발생에 의한 현상으로 튜닝 프로파일 만 적용 시의 한계로 판단된다.

CPU의 특정코어(Core #3)에 네트워크 프로세스를 전용으로 할당(Proc-Affinity, Int-Affinity)하여 사용하면 지터링(지연편차)과 최대지연 시간이 획기적으로 개선됨을 확인할 수 있다. 그러나 각각을 단독으로 적용 시 평균 전송시간은 network-latency 튜닝 프로파일 적용 시 보다 조금 크게 나타난다. 옵션에 대한 개별적용 결과로부터 통합옵션(3가지 옵션을 동시적용)을 적용하면 성능이 향상될 수 있음을 기대할 수 있는데 실제적으로도 통합옵션 적용 시 종합 성능이 향상되었음을 그림 6~9를 통해 확인할 수 있다.

통합옵션 적용 시 데이터크기/전송주기별 정량적인 전송성능을 표 4, 5에 나타내었다. 데이터크기가 커지면 전송지연이 비례적으로 증가하나 전송주기와는 관련성이 없음을 볼 수 있다.

표 4. 전송성능(1KHz 전송)

Table 4. Network performance(1KHz)

Size	64	256	512	768	1024	1518
Mean	31.8	37.6	44.6	50.7	58.1	70.5
Max.	84.4	95.6	90.3	102.7	110.4	181.3
Min.	23.0	30.2	38.7	41.5	48.3	61.0
Jitter	3.9	4.0	3.7	3.64	3.7	3.7

표 5. 전송성능(200Hz 전송)

Table 5. Network performance(200Hz)

Size	64	256	512	768	1024	1518
Mean	32.5	38.0	45.0	51.0	57.8	70.6
Max.	85.5	90.6	97.5	102.6	117.2	143.6
Min.	23.0	32.0	38.7	44.9	51.1	64.2
Jitter	4.6	3.9	3.9	3.7	4.1	3.9

이더넷으로 1회 전송 가능한 최대크기(1518 Byte)로 주기전송 시 평균전송시간은 70 $\mu$ s, 최대지연은 180 $\mu$ s 수준이며 지터링(편차)은 4 $\mu$ s 이내의 성능을 얻을 수 있었다. 일반적으로 실시간처리를 요구하는 제어 데이터들의 전송주기는 빠르나 크기가 작은 편이다. 실시간처리가 중요한 분야에는 1회 전송 데이터 크기를 64Byte(헤더를 제외한 사용자 데이터크기는 22Byte임)수준으로 제한하여 사용하면, 최대크기(1518 Byte) 전송의 경우보다 2배 이상의 전송성능 향상이 가능함을 확인하였다.

#### IV. 무인차량 네트워크(N:N) 성능시험

논문에서 소개하는 무인차량은 통제차량으로부터 수 Km 이상의 선단에서 원격 또는 자율주행으로 감시정찰, 전투임무 수행을 목적으로 하며, 2018년부터 2.5년 간 개발 및 평가를 완료하였다.

성능시험은 원격/자율주행, 감시, 사격 등을 중심으로 수행되었고, 시험 간 차량네트워크의 동작 안정성 확인은 기록장치 데이터 분석을 통하여 수행하였다. 그림 10은 개발된 무인차량시스템과 원격 사격시험 사진이다.

차량내부의 전자장치들은 이더넷을 백본으로 네트워크를 구성하였고, 각 노드를 구성하는 장치는 단위네트워크 시험에서 소개된 SBC, OS, 기술적 방법들을 탑재하였다. 각 노드에서 생성 데이터량은 확정적이며 정적라우팅 방식의 전송과 데이터충돌 방지 및 실시간 전송을 위하여 전이중 방식의 스위치 적용, 스위치버퍼조정, TCP/UDP 프로토콜의 선택활용, QoS 적용 등의 다양한 기법을 적용하여 표준 이더넷의 장점을 살리면서 데이터 전송의 불확실성을 최소화와 성능향상을 구현하였다.



그림 10. 무인차량형상 및 사격시험  
Fig. 10. Image of UGV and live fire test



또한, 단위네트워크 시험을 통해 확보한 통합업선을 동일하게 지상무인차량에 탑재하였다.

무인차량시스템에는 다수의 운용조종/전시 장치와 통제/감시/자율/항법/주행 장치들이 탑재되어 있고, 각각의 장치들은 상호간에 제어/상태신호를 주고 받으며 동작하는 N:N 통신시스템이다. 무인차량 성능시험 결과는 중요한 실시간 데이터가 가장 빈번하게 송수신되는 구간에 국한하여 소개하며 이와 관련된 네트워크 구성은 그림 11과 같다.

통제차량/무인차량 간 무선전송은 30/50Hz를 기본으로 하며, 무인차량내부에는 30/50/100/200Hz 등의 실시간 제어신호가 공존한다. 이외에도 10/5/1Hz 등의 주기신호와 다양한 비주기신호들이 있으나 논의에서 제외한다. 시스템내부의 제어신호 중 가장 빠른 실시간처리를 요구하는 분야는 사격제어루프이다. 통제차량에 탑승한 운용자가 감시장치로 부터 전송된 영상을 통해 표적을 조준하고 사격하면 자동으로 포탄이 표적에 명중할 수 있도록 무장의 조준선을 탄도값( $\theta_B$ ) 만큼 보정하는 기능으로 그림 12와 같은 구성을 가진다.

사격제어와 관련된 장치의 제어주기는 200Hz 로서 그림 13과 같이 5ms 프레임시간 동안 명령/센서 신호입력, 알고리즘처리 및 결과신호를 출력 할 수 있어야한다. 시스템에서 5ms의 프레임시간에 대한 활용은 알고리즘연산에 90%, 데이터 송수신에 10% 이하를 목표로 한다.

네트워크 성능평가는 장치들 간의 전송지연시간

을 측정하였다. 무인차량과 통제차량에는 그림 11과 같이 기록기가 장착되어 있으며, 이더넷으로 연결된 모든 장치들은 메시지의 송수신과 동시에 동일한 메시지와 시각정보를 기록기에 저장한다. 기록기 내부에는 그림 14와 같이 GPS 시각정보를 기반으로 정밀시간 생성장치가 내장되어 메시지 저장 시점에 타임스탬핑 하므로, 기록기 저장데이터를 활용하면 모든 장치 간 네트워킹성능을 분석할 수 있다.

네트워크 성능에 대한 시험결과를 그림 15와 표 6, 7에 나타내었다. 10분 동안 송수신된 50/100/200Hz 메시지에 대해 장치 간 총 전송지연시간을 그림 15와 표 6에 나타내었다.

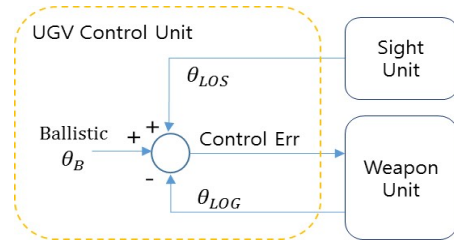


그림 12. 사격제어루프(200Hz)  
Fig. 12. Fire control loop(200Hz)

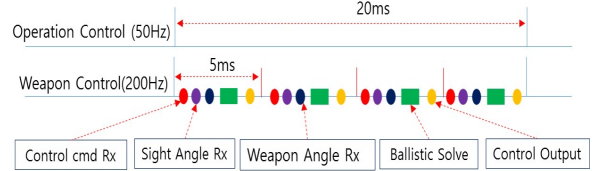


그림 13. 제어처리 프레임  
Fig. 13. Processing frame

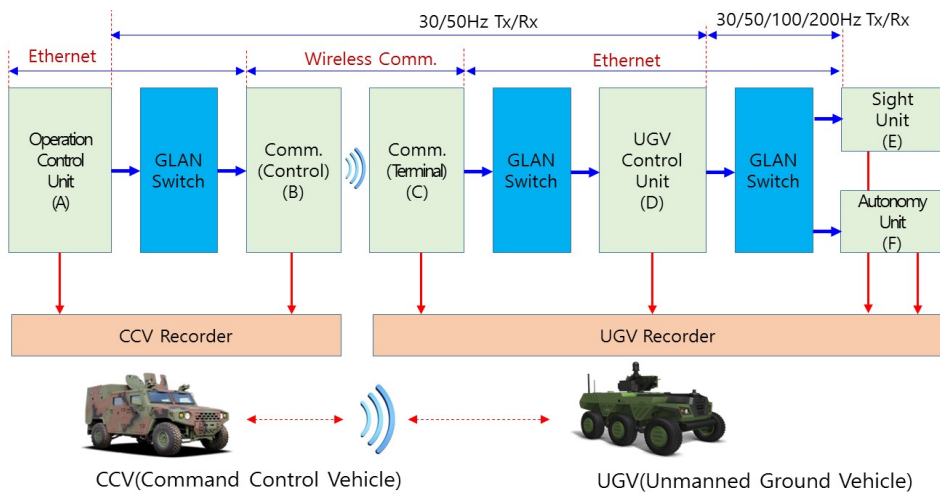


그림 11. 네트워크 시험구성  
Fig. 11. Test configuration of the network

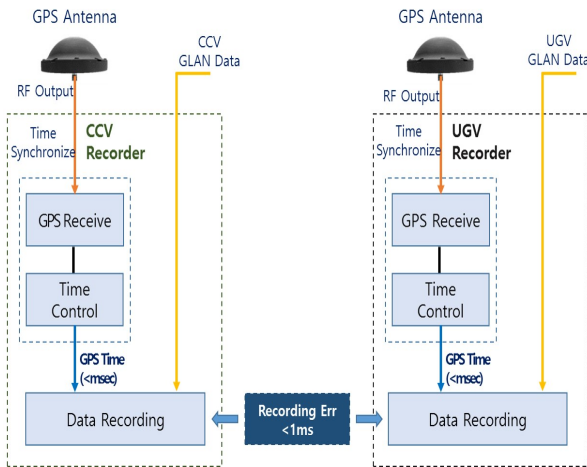


그림 14. 정밀시간 생성  
Fig. 14. Precision time generation

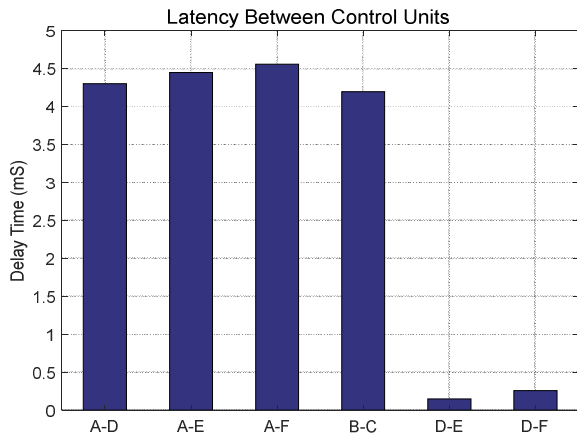


그림 15. 장치간 총 전송지연  
Fig. 15. Total latency between the units

표 6. 총 전송지연  
Table 6. Total latency

Type	A-D	A-E	A-F	B-C	D-E	D-F
Latency(ms)	4.29	4.45	4.56	4.20	0.15	0.26

표 7. 인터넷 전달지연  
Table 7. Ethernet latency

Type	A-D	A-E	A-F	B-C	D-E	D-F
Latency(ms)	0.09	0.25	0.36	-	0.15	0.26

조종장치(A), 통신장치(B)와 연결된 장치들은 무선구간을 통해 메시지 전송이 이루어지며, 총 지연 시간이 4.2~4.6ms 수준으로 비교적 크게 나타남을 볼 수 있다(장치 A~F 정의는 그림 11 참조).

이는 통신장치 간(B-C)의 무선구간에 의한 공통 지연시간 4.2ms 가 포함되기 때문이다. 논문에서 관

심의 대상이 되는 인터넷에 의한 지연성을 얻기 위해서 무선통신지연을 배제하면 표 7과 같이 장치간 전송성능 값을 얻을 수 있다.

200Hz 주기로 처리되는 사격제어 장치(D-E) 간의 지연은 0.15ms로서 메시지 전송시간이 프레임타임(5ms)의 3% 점유하므로 사격제어 처리에 문제가 없음을 알 수 있다. 기타 장치들은 100Hz 이하의 제 주기로 동작하며 최대 전송지연이 0.4ms 이하 이므로 무인차량의 실시간성은 보장된다고 볼 수 있다.

## V. 결 론

기존 군용차량에서 적용한 표준버서들은 전송용량 제한으로 대용량 영상신호와 제어/상태 정보를 동시에 전송하는 것이 불가능함에 따라 신호형태/종류 별로 다수개의 전용회선을 사용하고 있다.

본 논문에서는 군사용 지상무인시스템에 표준인터넷을 적용하여 대용량신호와 제어/상태 정보의 실시간전송 구현이 가능함을 시험을 통하여 입증하였다. 단위네트워크 시험에서는 실시간 OS 탑재 하에 CPU 코어 Shielding 및 Interrupt Affinity, Process Affinity, Tuning Profile 기능 적용에 따른 전송성능 분석을 통하여 안정적이며 최상의 성능확보를 위한 옵션설정 방법을 제시하였고, 결과를 실제 무인차량 시스템에 적용하였다. 무인차량시스템에는 정적라우팅 방식, 전이중(Full-duplex) 방식의 스위치 적용, 스위치버퍼조정, TCP/UDP 프로토콜의 선택활용, QoS 적용 등을 적용하여 데이터 전송의 불확실성을 최소화 하였으며 다양한 성능시험을 통해 차량 네트워크의 실시간성이 보장될 수 있음을 확인하였다.

네트워크 성능은 OS종류, Kernel, 탑재 플랫폼, 네트워크 컨트롤러, 스위치성능, 노드수, 데이터크기/발생빈도 등의 다양한 변수에 영향을 크게 받을 수 있으므로 응용체계에 적용을 위해서는 요구조건 분석을 통한 설계와 사전에 충분한 시험을 통한 입증 필요하다. 본 논문에서는 이와 같은 절차를 거쳐 무인차량시스템에 인터넷 기반의 실시간 네트워크 구현이 가능함을 입증하였다. 논문에서 소개된 결과는 향후, 인터넷을 기반으로 실시간 네트워크

설계를 필요로 하는 응용분야에서 참조 및 활용이 가능한 적용 사례가 될 것으로 기대된다.

### References

[1] D. J. Kim, D. W. Yang, Y. W. Lee, and T. K. Kang, "Real Time Networking Performance Implementation Based on Standard Ethernet", The Korea Institute of Military Science and Technology Symposium, Vol. 2019, pp. 1457-1458, Jun. 2019.

[2] P. Danielis et al., "Survey on Real-Time Communication Via Ethernet in Industrial Automation Environments", Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, pp. 1-8, Sep. 2014. <https://doi.org/10.1109/ETFA.2014.7005074>.

[3] T. P. Corrêa, L. Almeida, and E. B. Peña, "Hardware/Software Implementation Factors Influencing Ethernet Latency", 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), pp. 323-328, Sep. 2018. <https://doi.org/10.1109/INDIN.2018.8472002>.

[4] J. Sommer et al., "Ethernet - A Survey on its Fields of Application", in IEEE Communications Surveys & Tutorials, Vol. 12, No. 2, pp. 263-284, Apr. 2010. <https://doi.org/10.1109/SURV.2010.021110.00086>.

[5] M. Felser, "Real-Time Ethernet-Industry Prospective", in Proceedings of the IEEE, Vol. 93, No. 6, pp. 1118-1129, Jun. 2005. <https://doi.org/10.1109/JPROC.2005.849720>.

[6] J. Loeser and H. Haertig, "Low-latency hard Real-Time Communication Over Switched Ethernet", Proceedings. 16th Euromicro Conference on Real-Time Systems, Catania, Italy, pp. 13-22, Jul. 2004. <https://doi.org/10.1109/EMRTS.2004.1310992>.

[7] Si Gwan Kim, "Routing Protocols in VANET for Better Performance", Journal of KIIT, Vol. 19, No. 7, pp. 67-73, Jul. 2021. <https://doi.org/10.14801/jkiit.2021.19.7.67>.

[8] M. R. Park, M. H. Yoon, K. Y. Na, and D. W. Kim, "A Study on Application of Time-Triggered

Ethernet for Vehicle Network", The Journal of The Institute of Internet, Broadcasting and Communication (IIBC), Vol. 15, No. 6, pp. 79-88, Dec. 2015. <http://dx.doi.org/http://dx.doi.org/10.7236/JIIBC.2015.15.6.79>.

[9] RedHawk Linux User Guide (0898004-300), <https://redhawk.concurrent-rt.com/docs/>. [accessed: Sep. 14, 2021]

### 저자소개

김도종 (Dojong Kim)



1987년 2월 : 경북대학교  
전자공학과(공학석사)  
2001년 2월 : KAIST  
전기전자공학과(공학박사)  
2003년 ~ 현재 : 국방과학연구소  
연구원  
관심분야 : 로봇시스템, 네트워크,  
표적탐지/인지/추적

김남곤 (Namgon Kim)

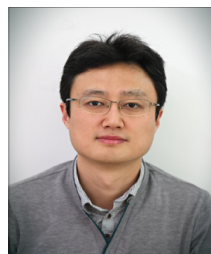


2006년 2월 : 광주과학기술원  
정보통신공학과(공학석사)  
2012년 8월 : 광주과학기술원  
정보통신공학과(공학박사)  
2012년 9월 ~ 2019년 12월 :  
(주)케이티 융합기술원 연구원  
2019년 ~ 현재 : 국방과학연구소

연구원

관심분야 : 네트워크가상화, In-Vehicle Network, 무인로봇

양동원 (DongWon Yang)



2001년 2월 : 서울대학교  
전기공학부(공학사)  
2003년 2월 : 서울대학교  
전기컴퓨터공학부(공학석사)  
2016년 2월 : KAIST  
전기전자공학부(공학박사)  
2003년 ~ 현재 : 국방과학연구소

연구원

관심분야 : 로봇시스템, 네트워크, 영상처리, 인공지능