

사물인터넷을 활용한 지능형원격제어시스템 구현

김동민*

Implementation of Intelligent Remote Control System based on Internet of Things

Dong Min Kim*

*Assistant Professor, Department of Internet of Things, Soonchunhyang University, Asan 31538, Korea

요약

원격제어 서버와 작동기가 무선네트워크를 통해 연결되어 동작하는 원격제어시스템은 그 편의성이 큰 만큼 잠재된 위험성도 크다. 신뢰할 수 없는 무선 채널로 인해 제어명령이 유실될 수 있기 때문이다. 지능형원격제어시스템은 작동기에 제어명령을 유추할 수 있는 기능을 추가하여 제어명령을 수신하지 못하는 경우에도 동작하도록 해주는 시스템이다. 본 논문에서는 지능형원격제어시스템 테스트베드를 구현하여 원격제어시스템에서 발생할 수 있는 문제 상황들을 실험을 통해 확인하고 지능형원격제어시스템이 문제를 해결하는 것을 검증하였다. 지능형원격제어시스템은 일반적인 원격제어시스템이 높은 통신오버헤드를 감수해야 달성할 수 있는 성능을 더 적은 통신오버헤드만으로 달성할 수 있음을 확인하였다.

ABSTRACT

The remote control system, in which the remote control server and the actuator are connected and operated through a wireless network, has a great potential risk as well as its convenience. The control commands can be lost because of unreliable wireless channels. The intelligent remote control system is a system that adds a function to infer a control command to the actuator to operate even if the control command is not received. In this paper, we implemented an intelligent remote control system testbed and confirmed the problems that could occur in the remote control system through experiments and verified that the intelligent remote control system solves the problem. The intelligent remote control system can achieve the performance that can be achieved when general remote control system has high communication overhead with less communication overhead.

키워드 : 사물인터넷, 지능형원격제어시스템, 무선네트워크, 테스트베드

Keywords : Internet of Things, Intelligent Remote Control System, Wireless Networks, Testbed

Received 26 February 2020, Revised 27 February 2020, Accepted 5 March 2020

* Corresponding Author Dong Min Kim (Email: dmk@sch.ac.kr, Tel.: +82-41-530-1535)

Assistant Professor, Department of Internet of Things, Soonchunhyang University, Asan 31538, Korea

Open Access <http://doi.org/10.6109/jkiice.2020.24.4.546>

print ISSN: 2234-4772 online ISSN: 2288-4165

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서론

사물인터넷의 발달로 인해 가정에서는 스마트홈, 사무실에서는 스마트오피스, 공장에서는 스마트팩토리가 도입되는 사례가 늘고 있다[1-5]. 이러한 변화의 핵심은 가정, 사무실, 공장 등에 무선네트워크를 통한 원격제어 시스템(Remote Control System)을 도입하는 것이다. 작동기(Actuator)가 위치한 곳에서 일일이 해주어야 했던 작업을 원격지에서 할 수 있으므로 편의성이 높아진다. 5G 통신이 도입되면 이러한 흐름은 더욱 가속화될 전망이다[6]. 무선네트워크를 활용한 원격제어시스템은 그 편의성이 큰 만큼 잠재된 위험성도 크다. 제어신호를 보내는 원격제어서버(Remote Control Server)와 제어신호를 받아 동작하는 작동기(Actuator) 사이의 통신이 원활하지 않은 경우에는 정보가 한 번에 전달되지 않을 수 있으며 재전송을 하면 전송지연이 발생할 수 있으며 전송 자체가 성공하지 않을 수도 있다. 이런 상황에서는 작동기가 원하는 대로 동작하지 않을 가능성이 크다. 실시간으로 제어신호를 받아서 동작해야 할 때에는 큰 문제가 된다. 이것을 해결하는 한 가지 방법은 작동기를 지능적으로 동작하도록 만드는 것이다. 즉, 작동기에 원격제어신호를 유추할 수 있는 기능을 추가하여 제어신호를 수신하지 못한 경우에도 지능적으로 동작하도록 만들어주는 것이다. 본 논문에서는 이러한 시스템을 지능형원격제어시스템(Intelligent Remote Control System)이라고 부르기로 하자. 본 논문에서는 원격제어시스템에서 발생할 수 있는 문제 상황들을 실험을 통해 확인하고 이를 해결하는 다양한 방법들의 성능을 검증할 수 있도록 지능형원격제어시스템 테스트베드를 구현하는데 초점을 둔다.

II. 지능형원격제어시스템 개요

일반적인 원격제어시스템은 명령을 내리는 원격제어 서버가 작동기의 상황을 파악하기 위하여 각종 정보를 수신한다. 이로부터 적합한 제어명령을 결정한 후, 무선네트워크를 통해 명령을 전송한다. 작동기는 이 명령을 수신하여 명령대로 동작한다. 동작에 의해서 주변 상황이 변하거나 상태가 변할 수 있기 때문에 변화된 상황을 센서(Sensor)를 통해 감지한다. 작동기 또는 센서는 이

정보를 원격제어서버에게 전송한다. 이 과정을 반복함으로써 원격제어시스템이 동작한다. 그림 1에 이 과정을 도식화하여 나타내었다.

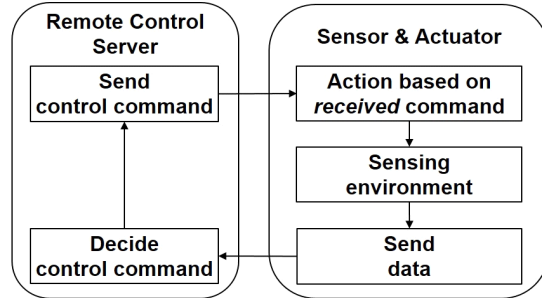


Fig. 1 Remote Control System

서론에서 언급한 바와 같이, 무선네트워크 환경에서 동작하는 일반적인 원격제어시스템은 통신 연결이 안정적이지 않으면 전체 시스템이 안정적으로 동작하지 못한다. 이를 해결하기 위한 기존의 연구들은 네트워크에 존재하는 전송지연의 특성을 분석하여 전송지연이 발생하기 전에 필요한 제어신호를 미리 전송하거나 앞으로 발생할 전송지연에 대비하여 모델예측제어(Model Predictive Control)를 수행하는 형태였다[7, 8].

최근에는 딥러닝 기술의 급속한 발달에 따라 기존의 시스템에 딥러닝 기술을 적용하여 시스템의 성능을 개선시키는 연구들이 등장하고 있다. 사물인터넷 분야에도 이러한 연구들이 활발하게 진행 중이다[9]. 더 나아가 에지컴퓨팅(Edge Computing)을 활용하여 시스템을 구성할 수 있다. 에지컴퓨팅은 센싱한 정보를 송신하는 기기나 전송 받은 정보를 다른 목적지로 보내주는 중계기와 같은 기기에게 계산능력을 부여하여 보다 효율적인 정보의 전달을 가능하게 하는 기술이다[10]. 최근에는 원격제어시스템 또는 네트워크제어시스템(Networked Control System)에 에지컴퓨팅을 적용하는 연구가 시도되고 있다[11]. 그러나 이 연구에서는 에지컴퓨팅 연산을 작동기 외부에서 수행하기 때문에 통신의 안정성에 영향을 받는 것은 기존과 같다. 본 논문에서 테스트베드를 구현한 지능형원격제어시스템은 일반적인 원격제어시스템이 가지고 있었던 무선네트워크의 안정성에 의존했던 문제점을 해결하기 위해 작동기가 지능적으로 동작하는 방식이다. 즉, 원격제어서버로부터 정보를 수신 받지 못한 경우 해당 명령을 추론하는 기능을 작동기

에 추가한 것이다. 그림 2에 지능형원격제어시스템의 동작 과정을 도식화하였다.

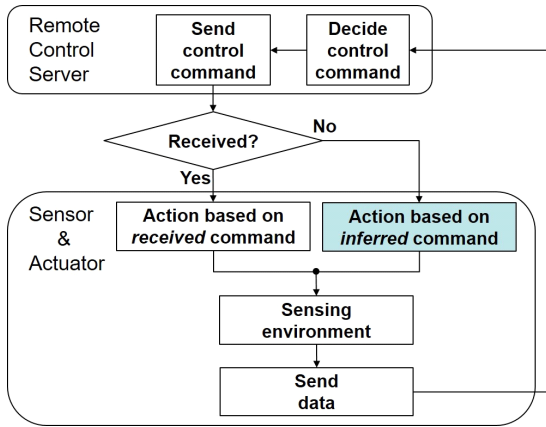


Fig. 2 Intelligent Remote Control System

지능형원격제어시스템의 원격제어서버는 일반적인 원격제어시스템의 원격제어서버와 같다. 명령을 받는 센서와 작동기의 동작에 차이점이 있다. 작동기는 제어 명령이 사전에 정의된 시간 간격(Time Interval)으로 수신될 것을 기대하고 있다. 이 시간 간격 안에 명령을 수신 받았으면 일반적인 원격제어시스템처럼 동작한다. 만약 명령을 수신 받지 못했다면 과거에 자신이 수행한 명령들과 그 때의 센싱정보들을 활용하여 현재의 상태와 센싱정보가 원격제어서버에게 주어졌을 때 어떤 명령이 전송되었는지 추론한 후 그 명령을 실행한다. 이때, 작동기는 원격제어서버가 어떤 원리에 의해서 명령을 내리는지 알지 못하며 자신이 전송한 센싱정보와 그에 대한 응답으로 수신 받은 제어명령만 알고 있다고 가정한다. 이 경우 명령의 형태에 따라 적합한 추론방식을 결정해야 한다. 추론의 정확도를 높이기 위해서는 제어 명령이 어떠한 유형인지 먼저 파악해야 한다. 예를 들어, 작동기의 특정 부분의 설정값을 연속된 값 중 하나로 설정하는 것인지 아니면 여러 개의 설정값 중 하나를 고르는 방식인지에 따라 연속된 값을 추정하는 방법과 여러 분류 중 하나를 선택하는 방법으로 나눌 수 있다. 다음 장에 설명할 테스트베드에서는 명령들을 분류하여 신경망을 통해 추론하였다. 추정의 정확도는 또한 사용할 데이터의 양, 추정 모델의 크기, 작동기가 사용하는 연산장치(Microprocessor)의 컴퓨팅 능력 등에 따라 달라질 것이다. 따라서 지능형원격제어시스템이 제

로 동작하기 위해서는 데이터를 축적시킬 시간이 필요하다. 또한 지능형원격제어시스템은 일반적인 원격제어시스템보다 작동기의 하드웨어 성능이 더 발달해야 한다. 다음 장에서 지능형원격제어시스템의 구현에 대해 살펴보겠다.

III. 지능형원격제어시스템 구현

지능형원격제어시스템을 실험하기 위하여 명령에 의해 동작하는 이동형로봇과 명령을 내리는 원격제어서버를 무선네트워크를 통해 연결하여 테스트베드를 구축하였다. 이 장에서는 테스트베드 구현의 세부사항을 설명한다.

3.1. 이동형로봇 구현

사물인터넷 장치인 라즈베리파이(Raspberry Pi) 3B+ 모델을 기반으로 동작하는 Waveshare Electronics에서 생산한 이동형로봇키트를 활용하여 이동형로봇을 제작하였다. 그림 3은 라즈베리파이가 장착된 이동형로봇의 모습이다.



Fig. 3 Mobile Robot Hardware

이동형로봇은 바닥에 5개의 적외선센서를 가지고 있으며 좌우 두 개의 모터가 바퀴에 연결되어 있어서 전후 좌우로 움직일 수 있다. 또한 라즈베리파이 카메라를 연결하여 이미지정보를 얻을 수 있다. 이와 같은 기능들을 활용하면 지도에 표시된 검은 색 선을 따라 이동하도록 제어할 수 있다.

그림 4는 이동형로봇의 동작 다이어그램이다. 이동형로봇은 통신모듈, 지능모듈, 모터모듈, 센싱모듈의 4가지 모듈로 구성되어 있다. 통신모듈은 제어명령을 수신하고 센싱정보를 송신하는 역할을 한다. 라즈베리파이의 무선랜기능을 활용한다. 수신받은 제어명령을 모터모듈에게 빠르게 전달하기 위하여 REST API를 사용하여 라즈베리파이의 GPIO와 연결하였다. REST API 구현은 WebIOPi를 이용하였다[12].

지능모듈은 지능형원격제어시스템의 핵심이라고 할 수 있다. 원격제어서버로부터 명령을 수신했으면 해당 명령을 저장한다. 수신 받은 명령은 과거에 전송했던 특정 센싱정보에 대응된다. 경우에 따라서는 전송지연이 발생하여 과거에 보냈던 센싱정보에 해당하는 명령이 뒤늦게 수신될 수도 있다. 정보의 나이(Age of Information)를 확인하기 위해 모든 센싱정보는 증가하는 일련번호를 가지고 있으며 제어명령도 대응되는 센싱정보의 일련번호를 가지고 있어야 한다. 실시간동작을 보장하기 위해 가장 최신의 센싱정보에 대응되는 명령을 수신했을 때만 해당 명령을 수행한다. 명령이 수신되지 않으면 지능모듈은 가장 최신의 센싱정보에 대응하는 명령을 추론한다. 본 논문에서 구현한 테스트베드는 좌회전, 직진, 우회전의 총 세 가지의 명령을 수신받는다. 따라서 명령추론은 3가지 명령 중 하나를 고르는 문제로 귀결된다. 과거 데이터가 충분히 쌓여있다면 깊은 신경망(Deep Neural Network)을 활용하여 추론하는 것이 좋은 성능을 낸다고 알려져 있다[13]. 라즈베리파이 상에서 동작하는 딥러닝 라이브러리인 Tensorflow와 Keras를 사용하여 깊은 신경망을 구축하였다. 10개의 입력노드로 깊은 신경망을 만들어 10자리의 이진수를 입력으로 삼았다. 각 입력노드는 적외선센서의 상태를 나타낸다. 적외선센서가 검은 색 길 위에 있으면 1, 아니면 0을 가지도록 했다. 한 번에 주고받는 센싱정보는 5개이지만 원격제어서버는 과거의 데이터를 함께 활용하여 제어명령을 결정할 수도 있다. 이동형로봇은 원격제어서버가 어떤 식으로 제어명령을 결정하는지 알지 못하지만 현재 센싱정보만으로 제어명령을 내린다고 추론하는 것보다 과거의 정보를 함께 고려하는 것이 더 합리적이다. 본 논문에서 구현한 테스트베드에서는 바로 이전 센싱정보까지 함께 고려하여 센싱정보를 추론하도록 구현하였기 때문에 10개의 입력노드를 가진다. 완전연결된(Fully Connected) 100개의 노드를 가진 2개의

Hidden Layer를 쌓았다. 이들의 활성화함수는 ReLU를 이용하였다. 출력층은 3개의 노드를 출력하며 Softmax를 이용하여 확률값을 나타내도록 했다. 출력층의 각 노드는 다른 명령을 나타낸다. 신경망의 구조 요약은 표 1에 나와 있다.

모터모듈은 지능모듈로부터 제어명령을 전달받아 좌회전, 직진, 우회전에 맞도록 기준에 설정된 값에 따라 두 개의 모터를 회전시킨다.

센싱모듈은 필요에 따라 적외선센서를 활성화시켜 센싱데이터를 갱신하거나 카메라를 켜서 전망이미지를 생성한다. 센싱정보는 통신모듈로 전달되어 원격제어 서버에게 전송된다.

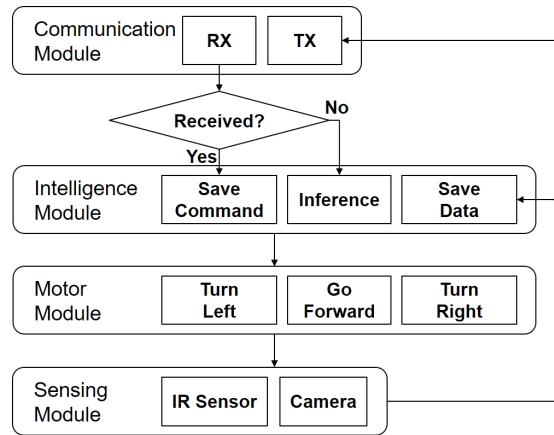


Fig. 4 Mobile Robot Operation Block Diagram

Table. 1 Neural Network Architecture

| Type | Filter Shape | Activation Function |
|---------|--------------|---------------------|
| FC1 | 10 × 100 | ReLU |
| FC2 | 100 × 100 | ReLU |
| Softmax | 3 | - |

3.2. 원격제어서버 구현

원격제어서버는 랩톱컴퓨터를 활용하였다. 작동기와 원격제어서버 사이의 무선네트워크는 WiFi를 활용하였다.

그림 5는 원격제어서버의 동작 다이어그램이다. 제어명령은 위에서 설명했듯이 REST API를 사용하여 전달하며 센싱데이터는 소켓통신을 통하여 전달받도록 하였다. 제어명령은 PID제어, 모델예측제어 등 다양한 제어기법을 활용하여 구현할 수 있다. 본 논문에서는 이

동형로봇으로부터 수신 받은 최신센싱정보와 바로 이전 센싱정보를 활용하여 이동형로봇이 가능한 한 검은 색 선을 벗어나지 않게 제어명령을 결정하도록 하였다. 다음 장에서 실험내용과 결과에 대해 살펴보겠다.

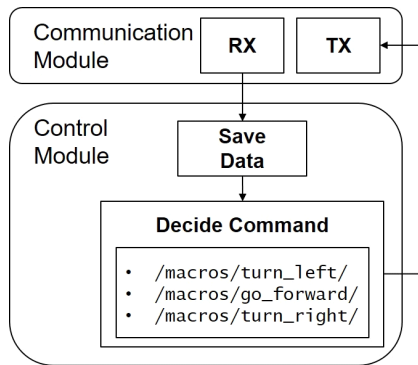


Fig. 5 Remote Control Server Operation Block Diagram

IV. 실험 수행 및 결과

4.1. 지능형원격제어시스템 실험 설명

지능형원격제어시스템을 실험하기 위하여 그림 6과 같이 테스트베드를 구성하였다. 그림에 보이는 것과 같은 간단한 지도 위에 이동형로봇을 위치시키고 경로를 따라가도록 하였다. 간단한 지도이지만 직진구간, 좌회전구간, 우회전구간, 교차로구간 등을 갖추고 있기 때문에 경로추적 성능을 검증할 수 있다.

이동형로봇이 경로를 잘 따라가면 지도를 한 바퀴 도는데 걸리는 시간(Lap Time)이 짧을 것이다. 또한 이동형로봇이 경로를 잘 따라가면 5개의 센서가 모두 길 위에서 벗어나는 경우(Offroad)가 적을 것이다. 이러한 점에 착안하여 실험에서는 Lap Time과 Offroad횟수를 측정하였다.

지능형원격제어시스템을 설계할 때 원격제어서버와 이동형로봇 사이에 얼마나 자주 통신을 해야 하는지 결정해야 한다. 시간간격을 너무 길게 하면 이동형로봇을 제어하기 어려워진다. 시간간격을 너무 짧게 하면 통신을 많이 하기 때문에 오버헤드가 커진다. 통신시간간격에 따른 성능을 측정하기 위하여 센싱정보를 보내는 간격을 0.05초, 0.075초, 0.1초, 0.125초, 0.15초로 설정하여 실험하였다.

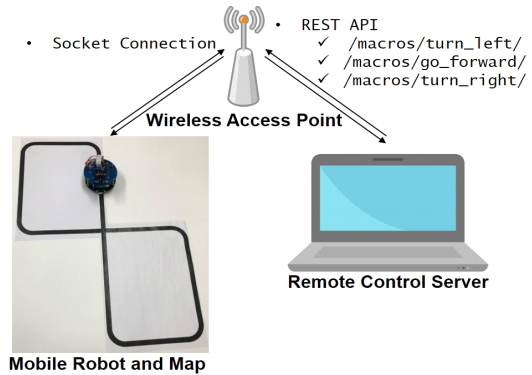


Fig. 6 Testbed Block Diagram

지능형원격제어시스템의 특징인 제어명령추론은 신경망에 의해서 동작하기 때문에 많은 데이터(Trajectory)를 가지고 있는 것이 좋을 것이다. 데이터양에 따른 추론 성능을 알아보기 위하여 특정 개수만큼 데이터를 수집했을 때까지 원격제어서버로부터 명령을 받고 그 이후에는 자체 추론기능만을 사용하여 이동형로봇을 제어하도록 하였다. 데이터는 5개의 적외선센서 상태에 따라 0 또는 1을 가지는 다섯 자리의 이진수이다. 이 값은 0~31의 값을 갖는 10진수로 변환하여 저장할 수 있다. 데이터는 총 256개, 512개, 768개, 1024개로 설정하여 실험하였다.

4.2. 지능형원격제어시스템 성능 측정 결과

그림 7은 Lap Time을 통신간격을 달리하여 측정된 결과를 그린 그래프이다. 통신간격이 짧을수록 Lap Time이 작아지는 것을 확인할 수 있다. 통신간격이 0.075초인 경우와 0.05초인 경우를 비교해보면 Lap

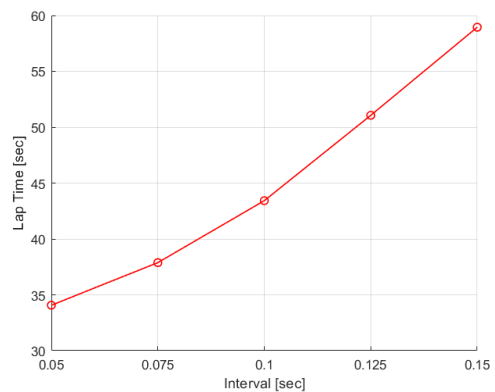


Fig. 7 Lap time as a function of interval

Time이 개선되는 정도는 그리 크지 않음을 알 수 있다. 통신간격을 줄여서 늘어난 통신오버헤드를 고려할 때 Lap Time의 개선정도가 크지 않다면 통신간격을 줄이는 것을 재고해볼 필요가 있다.

그림 8은 지도를 한바퀴 도는 동안 Offroad 횟수를 통신간격을 달리하여 측정한 결과를 그린 그래프이다. 통신간격이 짧을수록 Offroad 횟수가 줄어든다. 그림 7에서 확인한 것과 유사하게 통신간격이 작을 때 더 작게 만드는 것은 성능향상의 정도가 크지 않음을 알 수 있다.

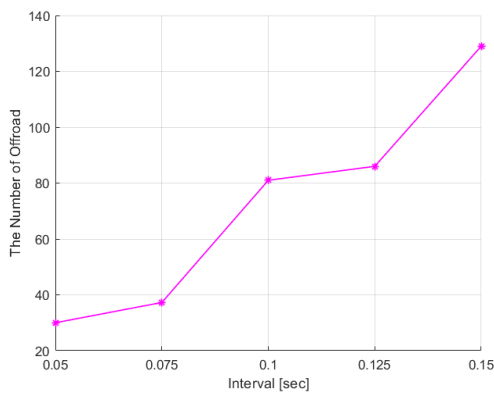


Fig. 8 The number of offroad as a function of interval

그림 9와 10은 원격제어서버와의 통신이 끊긴 이후 이동형로봇이 자체 추론을 통해서 경로추적을 하는 경우를 실험한 것이다. 통신간격이 0.05초, 0.1초, 0.15초인 경우도 같이 그렸다. 데이터양이 많을수록 Lap Time이 감소하는 것을 확인할 수 있다. 그림 10은 Offroad 횟수를 데이터양에 따라 그린 그래프이며 그림 9와 유사한 경향을 보인다. 두 그래프에서 주목할 점은 데이터양

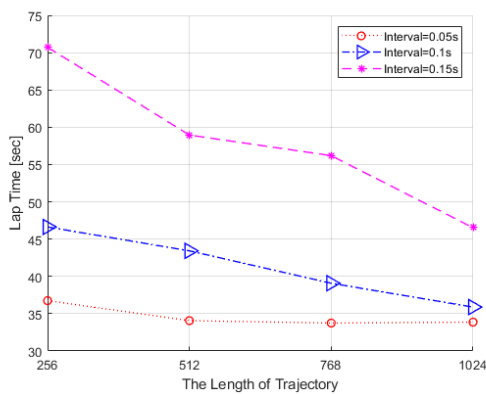


Fig. 9 Lap time as a function of the length of trajectory

이 1024개이고 통신간격이 0.15초인 경우와 데이터양이 256개이고 통신간격이 0.05초인 경우 성능이 비슷하거나 전자가 더 좋게 나타난다는 점이다. 즉, 더 큰 통신오버헤드를 감수해야 달성할 수 있는 성능을 데이터양을 늘림으로써 통신오버헤드를 늘리지 않고 달성할 수 있다.

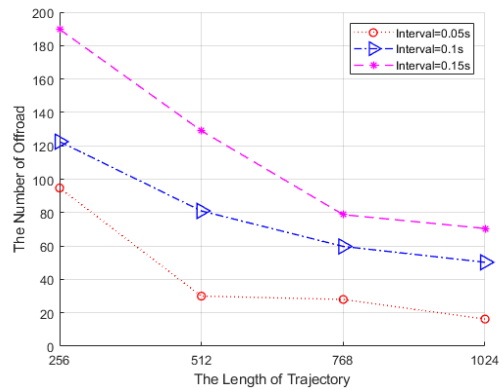


Fig. 10 The number of offroad as a function of the length of trajectory

V. 결론

본 논문에서는 지능형원격제어시스템의 유용성을 살펴보고 사물인터넷을 활용하여 이동형로봇을 원격으로 제어하는 테스트베드를 구현하였다. 구현한 테스트베드를 이용한 실험을 통해서 이동형로봇과 원격제어 서버 사이의 통신이 끊긴 상황에서도 지능형원격제어시스템은 정상적으로 동작하는 것을 확인하였다.

더 나아가 지능형원격제어시스템은 일반적인 원격제어시스템이 높은 통신오버헤드를 감수해야 달성할 수 있는 성능을 더 적은 통신오버헤드만으로도 달성할 수 있음을 확인하였다. 본 논문에서 제안한 기법의 성능에 대한 수학적 분석과 기존 알고리즘과의 비교분석을 포함하여 보다 실용적이고 복잡한 상황에 대해서도 지능형원격제어시스템을 구성하는 것이 앞으로의 주요 연구방향이다. 또한 라즈베리파이와 같이 연산능력이 떨어지는 사물인터넷 장비에서 딥러닝과 같은 무거운 연산을 수행하는 것은 버거울 수 있기 때문에 추론모델을 경량화하고 전력소모량을 측정하는 것이 지능형원격제어시스템의 발전을 위한 중요한 연구가 될 것이다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1G1A1100699). This work was supported by the Soonchunhyang University Research Fund.

REFERENCES

- [1] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190-1203, Apr. 2012.
- [2] J. Pan, R. Jain, and S. Paul, "A survey of energy efficiency in buildings and microgrids using networking technologies," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1709-1731, Jun. 2014.
- [3] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari, "Wireless communication for factory automation: An opportunity for LTE and 5G systems," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 36-43, Jun. 2016.
- [4] K. Lee, Y. Shin, Y. Lee, and S. Seol, "A study on user interface and control method of web-based remote control platform," *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, vol. 7, no. 6, pp. 827-837, Jun. 2017.
- [5] K. Cho, M. Jeon, and C. Oh, "Development of equipment control system based on DB access method for industrial IoT," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no. 6, pp. 1142-1147, Jun. 2016.
- [6] M. A. Lema, A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl, and M. Dohler, "Business case and technology analysis for 5G low latency applications," *IEEE Access*, vol. 5, pp. 5917-5935, Apr. 2017.
- [7] A. Seuret, L. Hetel, J. Daafouz, and K. H. Johansson, *Delays and Networked Control Systems*, Springer, 2014.
- [8] D. Jang, C. Y. Son, J. Yoo, H. J. Kim, and K. H. Johansson, "Efficient networked UAV control using event-triggered predictive control," in *Proceeding of the 8th IFAC Symposium on Mechatronic Systems MECHATRONICS 2019*, Vienna, Austria, 2019.
- [9] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: deep learning for the Internet of Things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96-101, Jan. 2018.
- [10] G. Premsankar, M. D. Francesco, and T. Taleb, "Edge computing for the Internet of Things: a case study," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1276-1284, Apr. 2018.
- [11] J.-B. Kim, D.-H. Kwon, Y.-G. Hong, H.-K. Lim, M.-S. Kim, and Y.-H. Han, "Deep Q-network based rotary inverted pendulum system and its monitoring on the EdgeX platform," in *Proceeding of the 1st International Conference on Artificial Intelligence in Information and Communication*, Okinawa, Japan, 2019.
- [12] WebIOPi [Internet]. Available: <https://webiopi.trouch.com/>.
- [13] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.



김동민(Dong Min Kim)

2014년 연세대학교 전기전자공학과, 공학박사
2014년 ~ 2018년, Aalborg University, Denmark, Postdoctoral Researcher
2018년 ~ 현재, 순천향대학교 사물인터넷학과, 조교수
※관심분야 : 사물인터넷, 네트워크제어시스템, 분산기계학습