

무인이동체 및 지상국 컴퓨터 간의 시간 정보 동기화를 위한 시스템 연구

이원석^{*} · 이운상^{*} · 송형규^{*†}

^{*†} 세종대학교 정보통신공학과

A Study on System for Synchronization of Multiple UAVs and Ground Control System

Won-Seok Lee^{*}, Woon-Sang Lee^{*} and Hyoung-Kyu Song^{*†}

^{*†} Sejong University, Information and Communication Engineering

ABSTRACT

In this paper, system that includes multiple unmanned aerial vehicles (UAVs) are considered. The vehicles are equipped with a mission computer for a specific mission and equipment. The mission equipment operates based on the time of mission computer. Also, data collected by flight computer and mission computer is saved with the time of each operating system. Generally, time offset between multiple computers always exists, though the computers are connected to the Internet. When the data collected by multiple computers is combined, the time offset causes damage on reliability of the combined data. Computers that connected to the Internet are synchronized by network time protocol (NTP). This paper proposes a system that the time of multiple mission computers are synchronized by the same NTP server to minimize the time offset. In the results of the measurement, the system time offset of multiple mission computer is maintained within 10ms from the system time of the server computer.

Key Words : Flight controller, Mission computer, Ground control system, Time synchronization, NTP, Time offset

1. 서 론

지상에서 동작하는 unmanned ground vehicle (UGV), 공중에서 동작하는 unmanned aerial vehicle (UAV), 해양에서 동작하는 unmanned marine vehicle (UMV) 등을 포함하는 무인이동체는 인간이 직접 탑승하지 않고 다양한 임무를 수행할 수 있기 때문에 오랜 기간 동안 유망한 분야로 고려되어 왔으며, 현재도 기술개발이 활발한 분야이다[1-4]. 무인이동체의 가장 중요한 특징은 인간이 직접 탑승하여 운용하지 않기 때문에 인력이 투입되기 어려운 장소에 투입되어 인간이 할 수 있는 임무들을 대신 수행할 수 있다는

것이다. 원전사고가 발생한 지역 그리고 대규모 건물 붕괴가 발생하여 추가적인 위협이 예상되는 장소들이 무인이동체가 투입되어 임무를 수행할 수 있는 대표적인 경우들이다. 현재 레저용 드론 시장의 70%를 점유하고 있는 중국의 DJI에 의해 레저용 드론이 주목을 받으며 무인이동체는 주로 레저용으로 인식되고 있다. 하지만 인류가 재해 및 재난 대처와 기술 실험 등을 효과적으로 수행하기 위해 무인이동체를 이용하는 것을 고려하면 레저용보다는 산업용 무인이동체 기술개발의 중요성이 더 높다고 볼 수 있다.

산업용 무인이동체는 레저용 무인이동체와는 다른 특징으로 인해 레저용 무인이동체에는 필요하지 않은 추가적인 기술 개발이 필요하다. 산업용 무인이동체는 인간이

[†]E-mail: songhk@sejong.ac.kr

직접 머무르기 힘든 곳에서 작업을 수행하기 때문에 비가시권에서 안정적으로 무인이동체를 제어하기 위한 원거리 무선 통신 및 무선 네트워크 기술이 필수적이다[4-6]. 현재 제어용 RF 통신 방식은 실제 현장에서 200~300m 까지만 안정적인 통신을 제공하는 수준이다. 즉 산업용 무인이동체를 위해서는 보다 원거리 통신이 가능한 LTE와 5G 이동통신 기술이 제어용 통신으로 고려될 필요가 있다[5-6]. 또한 이동통신 인프라가 충분하지 않은 환경에서는 무인이동체 간의 상호 무선 연결을 통한 중계 네트워크를 구축할 필요가 있다. 네트워크를 통해 비가시권의 무인이동체를 관제하기 위한 관제 시스템은 앞으로의 무인이동체 생태계를 위해서 가장 중요한 부분이다.

일반적으로 공중에서 운용되는 무인이동체의 기본적인 항법제어는 비행제어컴퓨터에서 수행하고, 임무장비는 임무컴퓨터에서 제어한다. 산업용 무인이동체는 정밀한 임무장비를 사용하기 때문에 두 컴퓨터의 시스템 시간에 차이가 발생하지 않도록 하는 것이 안정적인 임무 수행을 위해 필수적이다. 산업용 무인이동체는 다수의 무인이동체가 동시에 임무에 투입되는 경우가 많을 것으로 예측된다. 그리고 다수의 무인이동체가 동시에 작동하는 경우 각 무인이동체의 컴퓨터 간 정보의 오차는 전체 시스템에 더 큰 영향을 미칠 수 있다. 또한 다수의 무인이동체에서 습득한 데이터를 토대로 파노라마 영상과 같은 통합 데이터를 만드는 경우에도 동일 시간에 수집된 데이터가 통합되지 않는다면 데이터의 신뢰성을 떨어뜨린다. 일반적으로 인터넷에 연결된 컴퓨터들은 NTP를 통해 시스템 간의 시간 동기화를 맞추지만 현재의 NTP는 산업용 무인이동체 시스템에서 요구하는 동기화에 적합한 성능을 제공하지 못한다. 본 연구에서는 공중에서 운용되는 무인기들이 네트워크를 통해 동일한 관제시스템 하에서 동작할 때 각 무인기에 설치된 비행제어컴퓨터 및 임무컴퓨터 시간을 동기화하여 통합데이터의 신뢰성을 개선하기 위한 동기화 시스템을 제안한다.

2. 무인이동체 관제시스템 및 동기화

본 연구에서 고려하는 무인이동체 및 관제시스템은 Fig 1과 같이 나타낼 수 있다. 각 무인이동체는 공중에서 운용되며, 무인이동체 내부에는 기본적으로 항법 및 비행제어를 수행하기 위한 비행제어컴퓨터가 탑재된다. 그리고 임무를 가진 산업용 무인기의 경우 임무장비를 제어하기 위해 추가적으로 임무컴퓨터가 탑재된다. 임무컴퓨터는 비행제어컴퓨터에 의해 범용적인 계산을 수행할 수 있는 소형 컴퓨팅 보드에 해당한다. 최근에는 임무컴퓨터 역할을 수행할 수 있는 소형 컴퓨팅 보드가 사양에 따라

다양하게 공개되어 있는 상황이며, 대표적으로 교육용으로 인기가 많은 라즈베리파이 보드부터 좀 더 고사양의 컴퓨팅 능력을 요구하는 임무 수행을 위한 NVIDIA의 Jetson tx 보드 등이 있다.

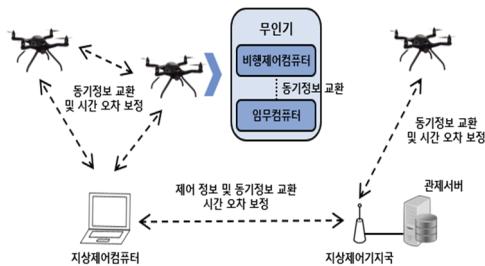


Fig. 1. 무인이동체 네트워크 및 통합 관제시스템.

Fig. 1에는 무인이동체들을 통합적으로 관제하는 관제서버가 존재한다. 관제서버는 지상제어기지국으로부터 인터넷을 통해 무인이동체에 대한 정보를 받아 관제임무를 수행한다. 관제서버는 unmanned aerial traffic management (UTM) 시스템에서 서버에 등록된 모든 무인이동체의 이동을 모니터링하며 특별한 상황이 발생했을 때 무인이동체를 직접 제어하는 운용자 혹은 소프트웨어에 경고를 하는 등 통합적인 교통관리 역할을 수행한다. 무인이동체의 통신은 다양한 방법으로 구현될 수 있다. Fig. 1의 경우는 무인이동체가 이동통신을 기반으로 관제시스템에 연결되는 경우를 보인다. 무인이동체 간의 통신 또한 이동통신을 이용하여 구현되거나 또는 고속교통수단 간의 통신을 위해 만들어진 IEEE 802.11p 등이 사용될 수 있다. Fig. 1에서는 비행제어컴퓨터와 임무컴퓨터를 모두 탑재하고 있는 무인이동체가 무인이동체 내부의 컴퓨터, 네트워크를 다른 무인이동체 그리고 관제시스템과 시간정보를 교환하며 시스템 시간을 조정하는 상황을 보인다.

2.1 비행제어 컴퓨터 및 시스템 시간

본 연구에서 대상으로 하는 무인이동체는 비행제어컴퓨터를 포함해 두 대 이상의 컴퓨터가 탑재된다. 각 컴퓨터는 컴퓨터의 목적에 맞는 운영체제가 설치되어 동작한다. 비행제어 컴퓨터의 경우 DJI의 드론과 같이 특정 회사에서 독자적으로 개발한 소프트웨어를 설치하는 경우와 현재 가장 잘 알려진 오픈소스 소프트웨어인 Ardupilot과 PX4가 설치되어 운용되는 경우가 있다. 오픈소스 오토파일럿들은 일반적으로 현재 가장 유명한 오픈소스 비행제어컴퓨터인 Pixhawk와 함께 동작하며 내부는 Fig. 2와 같은 구조로 되어있다.

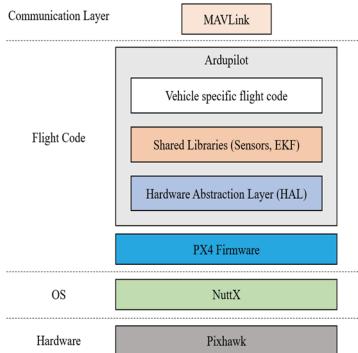


Fig. 2. 오픈소스 비행제어컴퓨터의 소프트웨어 구조.

오토파일럿 소프트웨어인 Ardupilot은 NuttX라고 불리는 실시간 OS위해 PX4 firmware가 설치된 환경에서 작동한다. Ardupilot은 NuttX와는 별도로 자신의 시스템 시간을 이용하여 데이터를 기록하는데 활용한다. 그리고 이러한 시스템 시간을 설정하는 방법은 사용자가 결정할 수 있도록 허용한다. 특별히 설정하지 않는다면 Ardupilot은 일반적으로 GPS를 이용하여 시스템 시간을 맞춘다.

2.2 NTP 기반 네트워크 동기화

NTP는 네트워크로 연결된 컴퓨터들의 시스템 시간을 동기화하기 위한 기술로 인터넷 프로토콜 가운데 하나이다. NTP는 UDP 패킷을 통해 메시지를 교환하는 서버, 클라이언트 기반의 프로토콜이며 일반적으로 20ms에서 50ms 사이의 정확도를 보인다고 한다. NTP는 Fig. 3과 같이 동작한다.

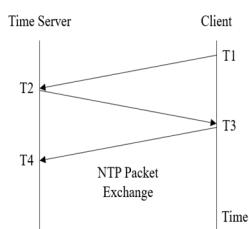


Fig. 3. NTP 기반 시간동기화를 위한 패킷 교환.

Fig. 3은 각 패킷 교환 시 패킷에 저장되어 전송되는 시간들을 나타낸다. 클라이언트는 서버의 시간과 오차가 존재한다고 가정하면 각 시간은 다음과 같이 나타낼 수 있다.

$$T_2 = T_s + T_d + T_o, \quad (1)$$

$$T_4 = T_s - T_o + T_d. \quad (2)$$

여기서 T_d 는 서버와 클라이언트 사이에서 패킷이 전송되는데 걸리는 시간을 의미한다. 여기서 클라이언트는 시간오차를 구하기 위해 (3)을 계산하여 시스템 시간에 반영한다.

$$\hat{T}_o = (T_2 - T_1 + T_3 - T_4). \quad (3)$$

양방향의 전송 지연이 같다고 가정하면 \hat{T}_o 는 두 시스템 시간의 차이를 나타낸다.

3. 제안 시스템

Fig. 4는 본 연구에서 제안하는 동기화 시스템 구성을 보인다. Fig. 4의 시스템은 비행제어컴퓨터와 임무컴퓨터가 탑재된 무인이동체와 통합관제서버 그리고 서버와 연결된 관제시스템 및 웹 기반의 운영시스템으로 구성된다.

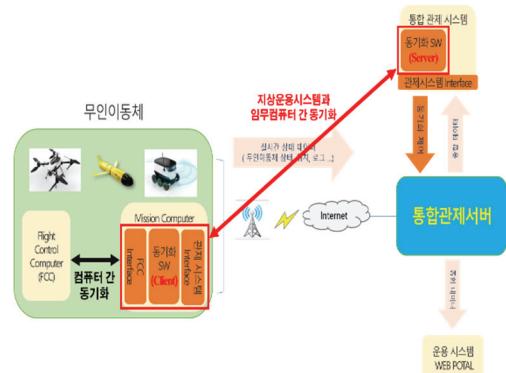


Fig. 4. 제안하는 동기화 시스템 구성.

제안하는 시스템은 다수의 무인이동체 간 시스템 시간 동기화를 위해서 무인이동체에 함께 탑재된 임무컴퓨터를 이용한다. 이를 위해 임무컴퓨터는 비행제어컴퓨터와 통신을 위한 인터페이스가 구축되어 있으며 MAVLink 프로토콜을 이용하여 통신이 가능하다고 가정한다. MAVLink는 2009년에 Lorenz Meier에 의해 소형 무인비행체 간의 통신을 위한 방법으로 처음 제안되었으며, 현재 오픈소스 오토파일럿 대부분은 MAVLink를 기반으로 동작한다. 비행제어컴퓨터가 오픈소스인 Ardupilot이 설치되어 동작하고 Ardupilot의 시스템 시간 설정방식을 임무컴퓨터의 시스템 시간을 받아와 사용하는 방식으로 설정한다면 두 컴퓨터 간 동기화는 주기적으로 수행될 것이다. 네트워크에 존재하는 임무컴퓨터 간의 시간동기화는 NTP를 이용한다. 기존의 NTP는 각 컴퓨터가 개별 서버와 동기화를 수행하는 경우가 발생하여 시간오차가 커질 수 있

다. 제안 시스템에서 모든 임무컴퓨터는 동일한 관제서버에 연결되어 관제시스템에 설치된 동일한 NTP서버를 기준으로 동기화를 수행하여 컴퓨터 간 시간 오차를 최대한 낮추도록 계획했다.

제안 시스템은 주기적으로 통합관제서버가 무인이동체의 임무컴퓨터들로부터 시스템 시간정보를 받아와 통계를 내고 결과 정보를 통합관제시스템과 웹 기반 운영시스템에 보내주는 방식으로 동작한다. 관제시스템의 운영자 판단 하에 시스템에서 요구하는 시간오차 값의 한계치가 정해지면 통합관제시스템은 통계치가 한계치를 넘어갈 때마다 NTP 서버를 이용하여 관제시스템에 연결된 모든 임무컴퓨터에 동기화를 위한 신호를 보낸다. 여기서 동기화 소프트웨어는 동기화를 위한 프로토콜로 NTP가 아닌 다른 프로토콜을 이용할 수 있다. 본 연구에서는 네트워크에서 보편적으로 많이 쓰이는 NTP를 이용하여 시스템을 구현했다.

3.1 소프트웨어 개발

소프트웨어 개발을 위해서 제안 시스템에서는 oneM2M 표준 기반의 통합관제서버를 이용하여 임무컴퓨터 간 시간정보를 모니터링하고 동기화 제어 기능을 구현했다[5]. 통합관제서버를 구현하기 위한 oneM2M 표준 소프트웨어 플랫폼은 오픈소스 프로젝트 OCEAN의 서버용 플랫폼 Mobius를 사용했으며 무인이동체의 임무컴퓨터에서는 Mobius와 데이터를 주고받기 위해 디바이스용 플랫폼인 &Cube를 이용했다. 임무컴퓨터의 시스템 시간 정보를 받아 출력하고 동기화 신호를 보내기 위한 통합관제시스템 소프트웨어는 Matlab의 graphic user interface (GUI)를 이용해 구현했다. Fig. 5는 개발한 시제품의 시스템 구조를 보인다.



Fig. 5. 개발 소프트웨어를 포함한 시스템 구성.

Fig. 5에서 oneM2M interface로 표현되는 구간에서는 oneM2M 표준 기반의 메시지 교환을 통해 임무컴퓨터로

부터 시간정보를 수집하거나 동기화를 포함한 무인이동체 제어 신호를 전송한다. Fig. 5에서 Mobius는 서버쪽의 소프트웨어와 HTTP, MQTT를 이용하여 통신한다. HTTP는 주로 필요에 의해 신호를 보내거나 특정 데이터를 가져오는 경우에 사용하며, MQTT는 publication과 subscription 기반의 프로토콜로 새로운 메시지가 발생했을 때 알림을 받는 방식으로 동작한다. 서버 소프트웨어는 무인이동체의 임무컴퓨터에서 Mobius로 전송되는 시간오차 정보를 모니터링하기 위해 MQTT 프로토콜을 사용한다. 동기화 소프트웨어의 서버는 Matlab을 이용하여, 클라이언트는 Python의 ntplib 모듈을 이용하여 개발했다. Fig. 6은 현재 개발이 진행 중인 소프트웨어 시제품 화면을 보인다.

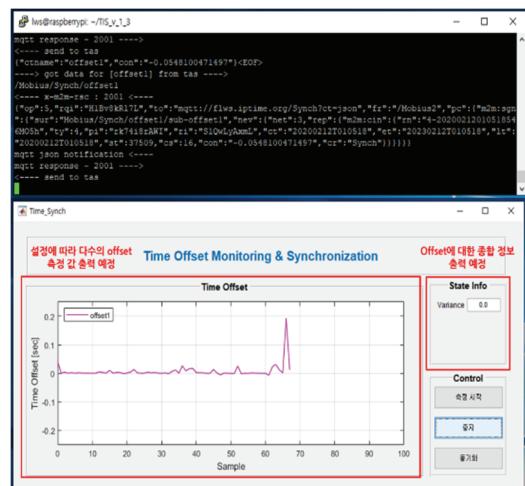


Fig. 6. 제안된 동기화 시스템을 위한 소프트웨어 시제품.

3.2 측정 결과

개발 소프트웨어를 이용하여 실제 무인이동체 운영에 많이 사용되는 Wi-Fi와 LTE 네트워크에서 측정을 수행했다. Wi-Fi 네트워크의 경우는 AP에 라즈베리파이 보드를 연결하여 서버를 설정한 후 시간오차를 측정했다. LTE의 경우는 두 가지 상황을 가정했다. 무인이동체가 LTE 네트워크에서 운영되는 경우 거리가 멀어지면 무인이동체들은 서로 다른 기지국과 통신을 하며 운용될 것이다. 이러한 다른 상황에서의 시간오차를 보기 위해 두 라즈베리파이를 동일 장소에 위치시켜 시간오차를 측정하는 방법과 200 미터 이상 떨어진 다른 건물에 라즈베리파이 보드들을 위치시켜 측정하는 방법으로 시간오차 측정을 진행했다. Fig. 7은 Wi-Fi 네트워크에서 측정한 결과를 보이며 Fig. 8과 9는 LTE 네트워크에서 두 가지 방법으로 측정한 결과를 보인다.

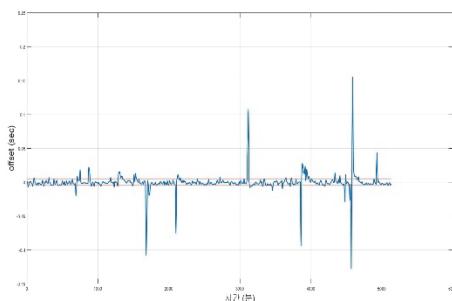


Fig. 7. Wi-Fi 네트워크에서의 동기화 성능 측정 결과.

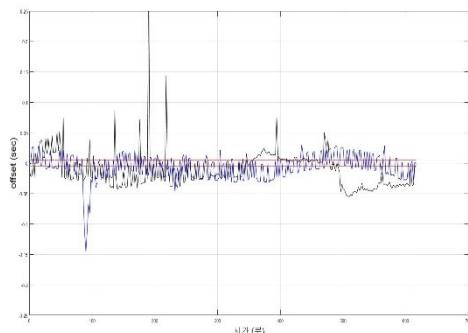


Fig. 8. LTE 네트워크에서의 동기화 성능 측정 결과
(동일 장소).

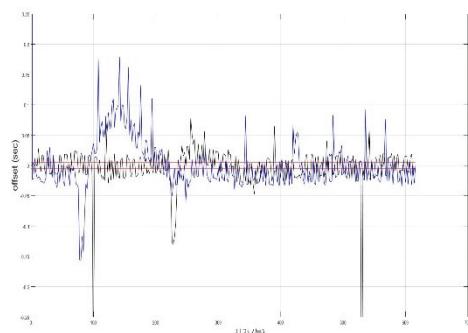


Fig. 9. LTE 네트워크에서의 동기화 성능 측정 결과
(다른 건물).

Fig 9에서 갑자기 시간오차가 크게 발생하는 구간은 측정 중간에 임의로 나타나는 상황임을 고려하면 두 LTE 네트워크에서의 측정결과는 개별 결과들로 비교할 때 큰 차이를 보이지 않는다. 하지만 두 상황에서 모두 동일한 관계서버를 동기화 서버로 잡고 있기 때문에 두 결과 모두에서 각 라즈베리파이의 시간오차가 유사한 범위 내에 존재하는 것을 볼 수 있다.

3.3 향후 계획

현재의 개발 시스템은 NTP만을 이용하여 임무컴퓨터 간의 시간오차를 유사한 구간에 위치시키기 위한 방법을 구현했다. 하지만 임무컴퓨터에서 동작하는 임무장비가 정밀한 운용이 필요한 장비임을 고려하면 시간 오차 성능을 10ms 이내의 구간에 위치시킬 필요가 있다. 향후에 추가적인 프로토콜을 적용하여 10ms 이내로 시간오차를 개선할 예정이며 각 비행제어 컴퓨터로 받아오는 GPS 정보를 이용하여 추가적인 성능개선을 진행할 예정이다.

4. 결 론

본 연구에서는 다수의 무인이동체가 존재하는 시스템에서 무인이동체의 임무컴퓨터 간 시스템 시간의 오차를 개선하기 위한 시간동기화 시스템을 제안한다. 본 연구의 시간동기화 시스템은 oneM2M 표준 기반 플랫폼을 이용하여 무인이동체의 임무컴퓨터 시간정보를 모니터링하고 임무컴퓨터에 설치된 소프트웨어를 작동시키는 방법을 사용했다. 시스템에 존재하는 모든 임무컴퓨터는 본 연구 과정에서 개발한 동기화 소프트웨어를 통해 관계서버와 연결된 동일한 NTP서버를 기준으로 동기화 기능이 동작하여 모든 임무컴퓨터의 시스템 시간이 유사한 오차범위 내에서 동작하도록 했다.

감사의 글

본 연구는 국토교통부/국토교통과학기술진흥원, 과학기술정보통신부, 산업통산자원부의 지원으로 수행되었음 (과제번호 1615011058).

참고문헌

- Culver, K. B., "From battlefield to newsroom: Ethical implications of drone technology in journalism," *Journal of mass media ethics*, vol. 29, no. 1, pp. 52-64, 2014.
- Alwateer M., Loke S. W., and Fernando, N., "Enabling Drone Services: Drone Crowdsourcing and Drone Scripting," *IEEE Access*, vol. 7, pp. 110035-110049, 2019.
- Choi, C. H., Jang, H. J., Lim, S. G., Lim, H. C., Cho, S. H., and Gaponov, I., "Automatic wireless drone charging station creating essential environment for continuous drone operation," *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)*, Ansan, pp. 132-136, 2016.
- Lee, J., "A Desing of Small Drone with Open Source Frame and Software," *Journal of the Semiconductor &*

- Display Technology, vol. 18, no. 2, pp. 78-79, 2019.
5. Duan, H., Wang, Z., Liu, Y., Li, X., Huang, H., and Zhao, H., “IWCA Algorithm for Clustered Drone Information Transmission Network,” 2018 5th International Conference on Soft Computing & Machine Intelligence (ISCFMI), Nairobi, Kenya, pp. 119-122, 2018.
 6. Choi, S., Sung, N., Park, J., Ahn, I., and Kim, J., “Enabling drone as a service: OneM2M-based UAV/drone management system,” 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, pp. 18-20, 2017.
 7. Shi, W. et al., “Multiple Drone-Cell Deployment Analyses and Optimization in Drone Assisted Radio Access Networks,” in IEEE Access, vol. 6, pp. 12518-12529, 2018.
-

접수일: 2020년 2월 13일, 심사일: 2020년 3월 11일,
제재확정일: 2020년 3월 18일