

Received March 25, 2020, accepted April 12, 2020, date of publication April 21, 2020, date of current version May 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2989276

Configuring RTK-GPS Architecture for System Redundancy in Multi-Drone Operations

INSEOP UM, SEONGJOON PARK¹, (Graduate Student Member, IEEE), HYEONG TAE KIM, AND HWANGNAM KIM¹, (Member, IEEE)

Electrical Engineering Department, Korea University, Seoul 02841, South Korea

Corresponding author: Hwangnam Kim (hkim@korea.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2020R1A2C1012389.

ABSTRACT The researches on Real Time Kinematic GPS (RTK-GPS) have been actively conducted to accurately estimate the location of drones, and there were many use cases applied to drones. However, such systems using RTK-GPS do not provide the protection against network instability, failure of the components that make up the RTK-GPS service. In other words, availability is not provided to ensure the stability of the RTK-GPS. In this paper, we propose an architecture enabling redundancy to improve the availability of RTK-GPS system by switching or parallelizing the GPS correction data sources. We also introduce how to propagate the correction messages through the drone network. After evaluating the system on empirical testbed, we validated that the correction messages propagated without delay, the message source was switched or multiplexed among several message sources through the proposed system when the network was unstable, and the switching time was short enough not to affect our propagation system. Consequently, we have confirmed availability through the proposed architecture of RTK-GPS redundancy.

INDEX TERMS GNSS, RTK-GPS, NTRIP, network RTK, vehicle, drone, UAV, UGV.

I. INTRODUCTION

Nowadays, almost of ourdoor devices receive their global location information through Global Navigation Satellite System (GNSS). To implement advanced unmanned systems like Unmanned Vehicle System (UVS) or Vehicle Ad-hoc Network (VANET), it is vital to recognize a precise location of all vehicles. There are a variety of navigation techniques to obtain these exact position values, but the most commonly used method is to measure the position through GNSS. Compared to the other technologies, GNSS has significant advantages in terms of the convenience and simplicity. However, measuring location in GNSS is not accurate enough to expand its usage due to some environmental and systemic factors. Therefore, depending on the situation, GNSS firmly limits the opportunity of the delicate robot applications, including the swarming robot applications.

Numerous studies on correcting the GNSS error are underway to obtain the accurate global location. Among these researches, one of the most commonly used and most accurate techniques is Real Time Kinematic GPS (RTK-GPS), and it

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Zhang¹.

is easier to use by deploying a Network RTK [1]. Network RTK System is known as a typical way of using RTK-GPS System. This system corrects the GNSS results using the error correction of the GNSS value calculated on the location of the many reference points, which commonly called *sources*. The reason why Network RTK system is commonly used is the higher position accuracy, shorter initialization time and better reliability than previous single-source RTK systems [2]. Most importantly, there was a notable improvement in coverage [3]. However, there have been no studies to increase availability through redundancy for RTK-GPS, and there was no analysis for it, so there was no guidance on how to configure the availability of the comprehensive navigation system. In this paper, we propose a basic design for RTK redundancy scheme in terms of cold standby, hot standby and multiplexing, analyze the availability of each design, and demonstrate the systems through actual implementation.

A. MOTIVATION

The disadvantage of Network RTK is that the further away it is from the mount that provides Radio Technical

Commission for Maritime Services (RTCM¹) format information, the larger the error is. In addition, the use of the Internet limits the accessibility of the system and makes it vulnerable to the cyber attacks which destabilizes the service's continuity. Moreover, since the mounting location is open in most cases without high-level emergency plan, so malicious physical attacks can easily occur. To the sum, there are serious defects in certain situations, for example, the situation where it is impossible to receive continuous and intact GPS signals, and the situation in which there is a lack of reference points for RTK calculation when the Internet is unavailable due to unstable or broken communication links. To avoid such situations, we need to build recovery schemes that can support RTK-GPS system to run robustly. To the best of our knowledge, there was no previous analysis of the availability of RTK-GPS for the use of RTCM messages. In this paper, we analyze the availability of RTK-GPS for the use of RTCM messages and propose cold-standby, hot-standby and parallel systems as redundancy systems.

B. CHALLENGE

Based on the above observation, we propose an architecture that enables redundancy to improve the availability of RTK-GPS, in which multiple sources that provides RTCM messages are employed and a switching system chooses one of the active ones. Specifically, we proposed a switching and parallel system to ensure that the RTK-GPS was run reliably without losing RTCM messages. To avoid losing messages, the switching system quickly switches from a failed RTCM message source to another available RTCM message source. The parallel system receives RTCM messages online from multiple stations and propagates the messages according to the desired time so that the system can always turn on RTK-GPS. Switching system can be categorized as cold-standby and hot-standby, and we showed that these standby systems and our novel parallel system were able to improve the availability of the RTK-GPS system in multi-vehicle system. Before analyzing the switching and parallel system, we found from the actual RTCM message collection that the mean time to failure of each single RTCM source can be fitted to the exponential distribution. Thus, we analyzed each structure through Markov chain and proved the improvement of the proposed system. Finally, to implement such a redundant system, we used the Here+ RTK base station as the local base station and the national open-access RTCM source as the network base station.

C. CONTRIBUTION

We confirmed that the RTCM message is propagated from the Ground Control Station (GCS) to the vehicles using our implementation. We also measured the switching time and calculated its availability. The failure rate for single RTCM

¹The proper name of the protocol for RTK and Distributed GPS (DGPS) is RTCM SC. 104. In this paper, we abbreviated as RTCM, for convenience and the visibility.

message source was between 0.59% and 35.32% when we set the timeout time as 1.01 second. For cold-standby, the availability was between 79.97% and 99.03%. Therefore, we saw a slight improvement in availability when using cold-standby. For hot-standby, availability ranged from 95.81% to 99.82%, indicating better availability compared to real-world and cold-standby. For parallel systems, we were able to ensure that the availability was 99.95% when paralleling two RTCM message sources and 99.98% when using more than three RTCM message sources. The details of the results are given in Section V.

The rest of this paper is organized as follows. Section II addresses the former researches from initiative global navigation system to the complicated RTK-GPS methods. Section III introduces our RTCM propagation system by means of switching and parallelizing method. Section IV shows our implementation of the proposed design and Section V discusses the experiment results. Finally, Section VI concludes the paper.

II. PRELIMINARIES

Before the body of this paper, we examine from the basic concepts of GNSS and RTK-GPS. We also review how the Network RTK can be used for precise localization.

A. GNSS

GNSS is a satellite navigation system that calculates the location and altitude around the globe using radio waves sent to the satellites orbiting the earth. The GNSS semantically includes GPS of USA, GLONASS of Russia, Beidou of China and Galileo of Europe. The system we propose depends on the function and performance of GPS receiver, but common GPS receivers use GPS, GLONASS and Beidou except Galileo. We address the features of GPS.

1) GPS

GPS was firstly developed for military purposes, but now it is freely used by civilians. GPS uses the World Geodetic System (WGS-84). Total 24 satellites are placed in six orbits to provide GNSS service, which is divided into Standard Positioning System (SPS) and Precision Positioning System (PPS). The signal for each service is transmitted by L1 (1575.42 MHz) and L2 (1227.6 MHz), respectively, [4]. They are 154 times and 120 times the basic frequency of 10.23MHz of the atomic oscillators (Cesium and Rubidium, respectively). These two signals are phase-modulated with C/A-code and P-code, where C/A-code is included in carrier L1 only, and P-code is included in both L1 and L2. Using these signals, GPS measures the delivery time of a code by comparing the code received from the satellite with the code generated by the code generator in the receiver at the same time. The position of the receiver is calculated using at least four distant measurements. However, the location values obtained by this method are not accurate due to the following reasons. First, there are errors by the ionosphere and the convection layer. The ionosphere layer exists between

50 and 200 kilometers above the ground and has electrically charged particles that cause bending when radio waves, such as carrier waves. The convection layer, which is regarded as be a common atmosphere, locates 8-16 kilometers above the ground and also refracts radio waves. Because their refraction rates are different, they have a greater impact on GPS errors.

There are many ways to solve this problem. The first method is to collect signals from the L1 and L2 carrier at the same time using a dual-frequency receiver [5]. Because the magnitude of the refraction is inversely proportional to the frequency, the index of the refraction can be calculated using the waves of different frequencies passing at the same time to correct the error. However, this method can only be applied to the ionosphere layer because the index of the convection layer's refraction is not related to the frequency. Also, the dual-frequency receiver is quite expensive. The second method uses one receiver, which uses the value "Mask Angle" to ignore the signals from satellites below any angle above the horizon [6]. The disadvantage of this method is that if the incident angle is too high, it may fall short of the minimum of four required satellites. Also, there is a multi path error. Especially, in areas where buildings are crowded, such as downtown areas, signals from the satellites are reflected to the building, so they are longer than the original length, causing errors. Next, there is a geometric error. This is indicated by the deployment of the satellites during measurement, if the satellite is moderately spread and distributed, will reduce the error and if the satellite is gathered on one side, the error increases. The even degree of satellite's deployment is called DOP (Dilution of Precision), and the error is calculated using PDOP (Positional DOP) among the DOP. Finally, there is an error by cryptic code. This is the error caused by the aforementioned C/A-code and P-code.

B. RTK GPS

RTK GPS is a GPS that generates GPS correction data using terrestrial base station, which already knows its location, and passes them to the rover to correct GPS values to obtain more accurate location values. As aforementioned before, GPS calculates the distance using a carrier waveform phase-modulated with P and C/A values and obtains the location. The correction data is calculated through the comparing the carrier phase, and transmits using data format protocol called RTCM.

- **RTCM:** Radio Technical Commission for Maritime Services is a committee that plays the role of defining the differential data link for real-time differential calibration of the GNSS rover receiver [7]. RTK-GPS system sends the necessary information for location correction of GNSS using the standard RTCM specification. Standards are largely categorized to `version2` and `version3`, and the details of both versions can be found in [7]. Currently, `version2` is rarely used and `version3` is mainly adopted. Therefore, we used `version3` for implementation.

- **RTCM propagation:** In general, to use RTK-GPS in drone operations, telemetry is largely used. To be more specific, first of all, the installed base station must be connected to the GCS with the platform such as QGroundControl installed. Then, the base station uses the datalink to stream RTCM correction data to the drone through the flight controller. Telemetry is most commonly used one in drone operation because it supports the transmission of MAVLink messages.

1) NETWORK RTK

This system allows the user to receive and use GPS correction data information from the nearest mount, which means a reference station, if the Internet is connected, without the need to install the receiver at the proximity of rover (the system that relies on RTK-GPS for positioning) or the communication equipment. When providing data, it uses the format Networked Transport of RTCM via Internet Protocol (NTRIP) [8]. Network RTK consists of NTRIP Server, NTRIP Caster, and NTRIP Client. The NTRIP Server receives data generated from the GPS reference station and provides the NTRIP Caster with continuous GPS data. The NTRIP Caster functions as an HTTP server, and configures all of the available NTRIP sources and defines their source IDs. The NTRIP Client selects NTRIP source according to the source table information provided by the NTRIP Caster and receives correction data from the NTRIP Caster by entering the ID and password. Network RTK has a number of techniques, including MultiRef, Virtual Reference Station (VRS), Flächen-Korrektur-Paramete (FKP), and Master-Auxiliary Corrections (MAC) [9].

In South Korea, National Geographic Information Services is building a nationwide RTK service infrastructure to provide network RTK services, and the Seoul government is building its own infrastructure. VRS, FKP service can be used by using National Geographical Information Service, and GNSS-data integration-center can be used to use GNSS information of each mount. The aforementioned VRS and the FKP use modelling performed by proprietary network software. The fact that proprietary information is transmitted means that the corrections are not standard, and therefore it is biased toward a particular product or brand. In addition, the VRS and the FKP method do not conform to the philosophy of RTCM's standard format because they do not use raw data as defined by the RTCM but contained the modelled data [10].

Studies has been conducted in various ways to cope with the situation where the aforementioned RTK-GPS system is intimately threatened. To begin with, there was a research that mounts three receivers sequentially so makes RTK-GPS receiver to have the high probability of getting unwounded GPS signal despite of unfavorable environmental conditions [11]. Since it has three receivers in a row, the receivers can cooperate with each other and come up with precisely calculated results even though the received GPS signal was corrupted due to multipath effect. However, this approach has a limitation that it is only focusing on collecting a precise GPS

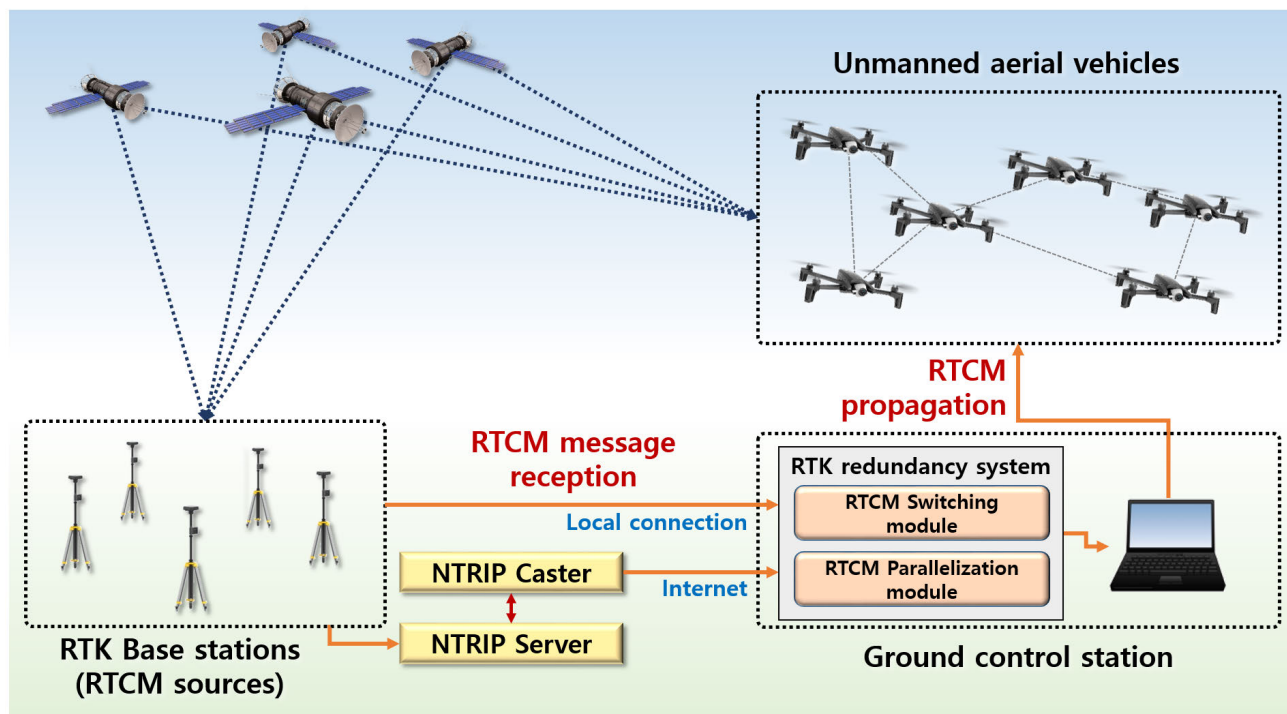


FIGURE 1. RTCM propagation system design.

to the receiver stage, while it cannot resist any problem in network latency. Next, there is a standalone RTK method that calculates the RTK-GPS data using the observation values stored in the receiver for dead spots [12]. It can generate the continuous RTK-GPS data by reusing the previously stored GPS values even though the system encounter the dead spot. But this study also has a limitation that it only considers the receiver stage. It is important to consider and improve not only receiver end but also transmission end to make whole RTK system durable. Furthermore, there is a study that dynamically regulates the data transmission rate in transmission end of RTK-GPS positioning data [13]. This approach can improve the system’s efficiency and economic feasibility by controlling the packet-switching communication with the consideration of environmental parameters. This system is capable of performing successful RTK-GPS transmission by flexibly adjusting the data rate in a threatened environment. Unlike the other studies, it improves the RTK-GPS system’s stability in transmission stage. However, there is a limitation in that it does not suggest a system solution in the situation that the RTK-GPS data is not completely arrived. As such, existing studies have limitations and this has become a motivation for us to propose our architecture.

III. RTCM PROPAGATION ARCHITECTURE

We propose a novel architecture to provide RTK-GPS more reliably for devices and users that need precise positioning. In order to show the performance of architecture we propose in the most dynamic and difficult situation, we select multiple

drones as target devices. There are many attempts to use drones for various applications, however, such attempts are almostly infeasible if it is not reliable to estimate the exact positions of the drones. The key reason is the high mobility of drones in large space, particularly in 3D space.

Our system architecture is shown in Fig. 1. It is composed of the base stations, NTRIP Servers&Casters, drones and a Ground Control Station (GCS). The GCS receives the information about the correction data from one or more base stations and NTRIP Caster. Then, the GCS periodically transmits the data to drones over the wireless network.

A. RTCM PROPAGATION

The main differences from the previous designs are the redundancy system and the connection method among drones while use RTK-GPS. As discussed in Section II, base stations use *telemetry* when they transmit RTCM messages to drones. Therefore, telemetry was inconvenient for sending RTCM messages because only one-to-one connections could be established. Moreover, if drones desire to receive RTCM messages using NTRIP, each of them has to be connected to the Internet and uses the NTRIP Client to connect to the NTRIP Server/Caster.

To resolve this inconvenience, our system configures a wireless ad hoc network to connect drones with GCS [14] and propagate RTCM messages. This approach has the advantage that drones do not need to be connected to Internet, since it receives the correction data through the local network. Especially, the system can verify the generated RTCM messages

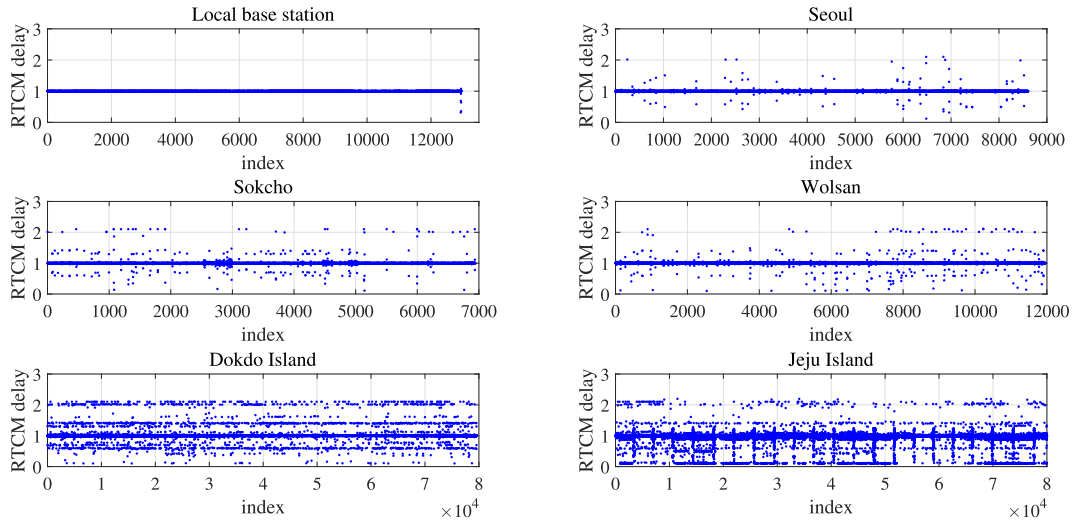


FIGURE 2. Inter-delay between RTCM messages from multiple RTK sources.

whether they are properly received. In addition, it can send the RTCM to the connected vehicles whatever kinds of the GPS receivers used. This feature leads to the breakthrough to improve the availability of the RTK-GPS of the drones, by increasing the number of the RTCM message sources connected to the GCS.

Such drone network based RTCM propagation design can experience the instability of the RTK-GPS navigation, due to the poor reception of the RTCM messages. We address two cases of the RTK-GPS malfunctioning, the one is the RTCM message delay, and the other is the RTCM source deactivation (by GPS denial, network disconnection, and so on). The former case occasionally happens due to the network delay between the RTCM source and the GCS, which could be larger in farther distance or the wireless connection. When RTCM messages are delayed, the drones use old correction data to correct GPS errors, so errors can accumulate during mission execution. We investigated the failure to the RTCM arrival time limit at Table. 3, and this delay occurs in 3 minutes at most. The latter case can happen due to the network disconnection between the RTCM source and the GCS, or the physical disruption to the source. If the RTCM source is down, the GCS cannot send RTCM message, and the drones experience the same error as GPS. To address this uncertainty in the provision of RTCM messages, we propose several types of redundant systems for RTCM reception, which ensures robust delivery of RTCM messages when one of the RTCS bases is disconnected due to network instability. The following subsections address the redundancy systems we propose.

B. RTCM REDUNDANCY SYSTEM DESIGN

The redundancy system is designed to ensure the availability while use RTK-GPS. In this system, the GCS receives the RTCM messages from one or more local base stations and the NTRIP Servers/Casters. If the reception status of the RTCM messages becomes poor in the existing connection,

due to malicious attacks, unstable Internet connection or GPS denied situations, then the system switches to another connection. For instance, if a connection of a local base station becomes unstable, the system switches the RTCM message source to another local base station or one of the NTRIP servers. On the other hand, if a connection to a NTRIP Server becomes unstable, then the system switches the RTCM source to another NTRIP server or one of the local base stations. We subdivided the method of this redundancy system in three ways: cold-standby (Section III-B.2) and hot-standby (Section III-B.3), and the parallel system (Section III-B.5).

1) ANALYZING INTER-DELAY DISTRIBUTION OF RTCM MESSAGES

We made a hypothesis for subsequent analyses that the average time to failure and the time to repair of active and standby units are exponentially distributed. To justify the hypothesis, we received RTCM messages from several RTCM message sources, and observed the inter-delay of the arrival times, plotted as Fig. 2. Then, we plotted a PDF of $\{t'_i | t_i > T\}$ using the measurements in Fig. 2, which is shown in Fig. 3, and tried to fit the PDF with various mathematical functions and finally we came to conclusion that exponential distribution is well fit for it with respect to Least Mean Square Error (LMSE). In other words, the distribution of t_i can be estimated to the exponential distribution, expressed to $f(x; \lambda) = \lambda e^{-\lambda x}$. Based on this, we modeled our proposed methods using Markov chain.

2) COLD-STANDBY

Cold-standby consists of one master system and several slave systems, and the slave systems are not in operation. After the instability of the master system is detected, it activates one of the available slave systems and continues the propagation. The probability of instability detection of master system is called failure detection probability, and is shown as Table 3

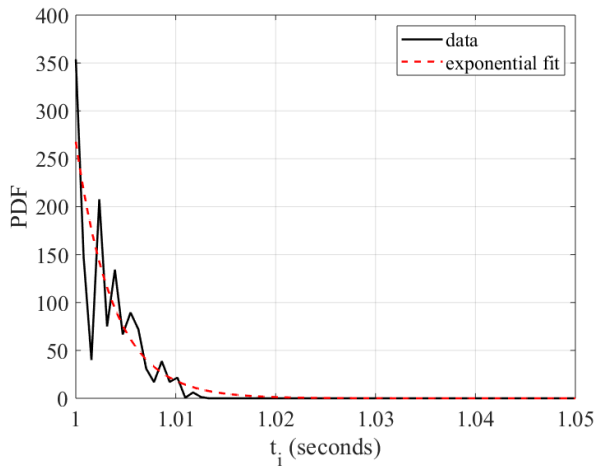


FIGURE 3. Probability distribution function of t_i and fitting result.

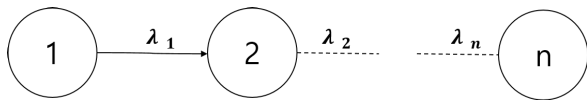


FIGURE 4. Markov process of cold-standby.

in Section V. The advantage of the cold-standby is relatively low resource requirement, because only one connection runs in a whole operation.

We modeled the Markov process of the cold-standby method. Each system is memoryless, so there is no return the previous state. Note that this method switches the malfunctioning master system to one of available slave systems, so we assume that the complete failure would not happen. That is, the probability of switching to a system that is already broken is zero. The Markov chain of the cold-standby is shown in Fig. 4, where the states mean switching of the master system.

The probability that a failure occurs in each state and changes to another state, that is, the *failure rate*, is denoted as λ . The failure rate may vary slightly depending on the network environment of each system, so we express them as $\lambda_1, \lambda_2 \dots, \lambda_n$. If the failure rates in each state are $\lambda_1, \lambda_2 \dots, \lambda_n$,

$$f_1(t) = \lambda e^{-\lambda_1 t}, f_2(t) = \lambda e^{-\lambda_2 t}, \dots, f_n(t) = \lambda e^{-\lambda_n t}. \quad (1)$$

Using the n -stage erlang distribution, it is expressed as

$$f(t) = \frac{(\lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_n) t^{n-1} e^{-\lambda_i t}}{(n-1)!} \quad (2)$$

$$F(t) = 1 - \sum_{i=1}^n \frac{(\lambda_i! t^i)}{i!} e^{-\lambda_i t}. \quad (3)$$

We implemented the cold-standby in Section IV-B and evaluated the availability in Section V-B.

3) HOT-STANDBY

Hot-standby operates the master and the slave systems together, but uses the data from the master system only. However, since the slave systems are already in operation,

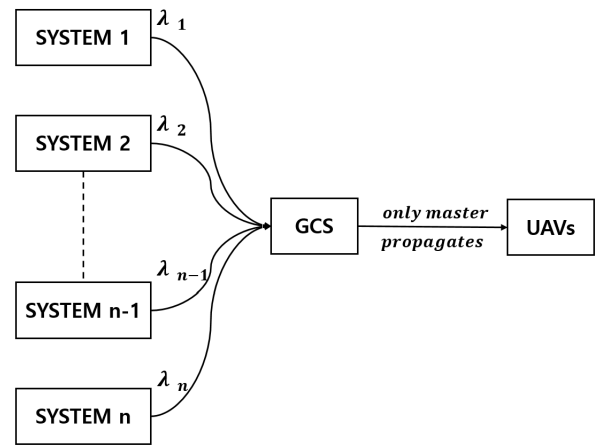


FIGURE 5. Hot-standby operation model.

TABLE 1. The state of Markov process in hot-standby setup.

State index	status
O	At least one works
F	All fail

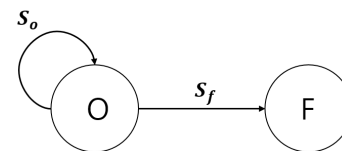


FIGURE 6. Markov process of hot-standby.

they can be switched as soon as the failure is detected. As shown in Fig. 5, all systems are running and connected to the GCS. But, only the master system's data is propagated to drones. The switching time takes shorter than cold-standby, but it has some disadvantages in resources such as memory and network traffic, since it always operates multiple systems.

Our proposed hot-standby has two states as shown in Table 1. The state change of this markov chain is shown in Fig. 6, and the probabilities of state change are described as

$$f_{S_o}(t) = 1 - e^{-\sum_{i=1}^n \lambda_i t} \quad (4)$$

$$f_{S_f}(t) = e^{-\sum_{i=1}^n \lambda_i t}. \quad (5)$$

Eq. (4) indicates the probability of receiving correction data from at the master system, and Eq. (5) refers to the probability that the all RTCM message sources are failed or broken.

4) ANALYSIS OF AVAILABILITY

To analyze the availability of system that consists of cold-standby components or hot-standby components, we derive the Mean Time To Failure (MTTF) of each method by

$$MTTF = \frac{\tau}{\lambda} \quad (6)$$

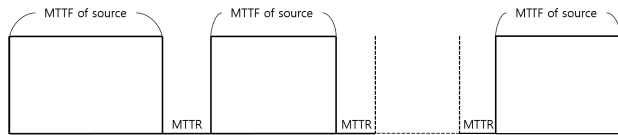


FIGURE 7. The operation according to switching system.

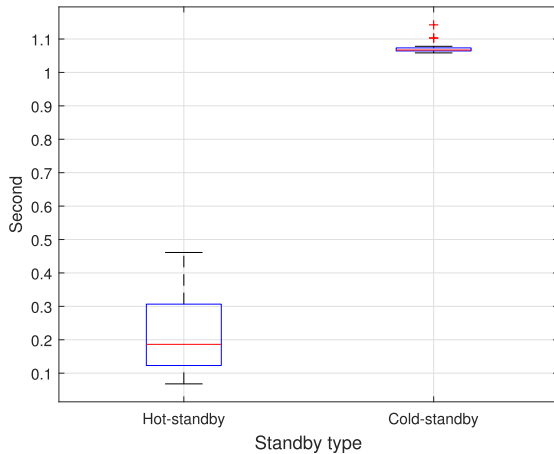


FIGURE 8. The switching time according to standby type.

where τ refers to the RTCM message reception interval. Also, Mean Time To Repair (MTTR) can be obtained experimentally through the practical switching system:

$$MTTR = t_d + t_n \tag{7}$$

where t_d is the failure detection time and t_n is the network delay. This system use NTRIP servers, so it has network delay (t_n) that means fail-over delay. Because, this delay is requisitely included in the system recovery process. From Eqs. (6) and (7), we can derive the availability of each method, expressed to

$$Availability = \frac{\sum_{i=1}^n MTTF \text{ of source } i}{\sum_{i=1}^n MTTF \text{ of source } i + (n - 1) \cdot MTTR} \tag{8}$$

Remind that the system using RTCM message sources among installed base stations and NTRIP servers can have different MTTFs.

The reason why MTTR is used in Eq. (8) is because it is large enough to affect the case of cold-standby. However, in the hot-standby’s case, MTTR is so small that it cannot affect the availability. Note that the cold-standby’s MTTR is relatively large because the slave systems are not in the operating state. To show the clear difference between the hot-standby and the cold-standby, we plotted the distribution of the switching times of each method, shown in Fig. 8. In Fig. 8 shows that the switching time of the hot-standby has the minimum value of 0.0680 second, the maximum value of 0.4612 second, and the median value of 0.1863 second. In the case of the cold-standby, it has the minimum value of 1.0585 second, the maximum value of 1.1424 second,

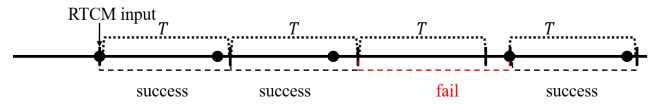


FIGURE 9. RTCM parallel system design.

and the median value of 1.0670 second. The deviation of the hot-standby’s switching time is greater than the one of the cold-standby, because the hot-standby is in receiving the RTCM message data from the slave system (RTCM message source), and the switching time includes a part of the inter delay of the slave’s RTCM messages, in addition to the master system’s reception failure detection time. That is, when the system detects the reception failure, the switching time changes depending on the remaining arrival time of the RTCM message at the slave system. On the other hand, in the case of the cold-standby, we can verify that that the deviation of the switching time is not significant because the slave system becomes active after the reception had detected. Also, comparing to the hot-standby, we can see that the switching time is about 0.9 seconds faster than the cold-standby, which is because of the time accessing to the NTRIP Caster at first and receiving the RTCM message data through the Internet. Considering that the switching time of the hot-standby and the cold-standby are 0.0680 and 1.0670, respectively, we know that the hot-standby will certainly not violate the delay threshold of the RTK-GPS, so it can increase the availability of drone’s RTK-GPS system. For cold-standby, in some situations where the computational and the network resource are highly restricted, it can be selected as an energy-efficient solution of the system redundancy since its switching time is also not fatal.

5) PARALLEL SYSTEM

In addition to the switching system, there might be a possibility to choose a RTCM message if multiple RTCM sources are available. We designed a novel parallel system architecture that the system periodically selects one of the received RTCM messages and propagates to the multi-drone network. This system receives the RTCM messages from the desired RTCM message sources, customizes it and transmits it, unlike the aforementioned switching system that does not receive RTCM messages from each RTCM message source at once.

The graphical representation of the parallel system is shown in Fig. 9, where black dots represent the arrival time of the RTCM messages. The system initiates when a first RTCM message comes, and runs a timer that measures every T interval, where T refers to the length of a time slot. For each time slot, the system gathers the RTCM messages from local base stations or NTRIP sources, and picks a message from these messages following the certain rules, and then propagates the messages to the network. If a RTCM message propagates to the network within T , the system state is regarded as success. On the other hand, any RTCM message has not arrived until

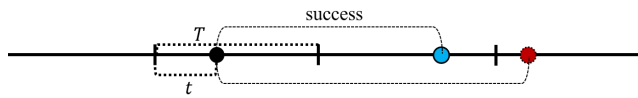


FIGURE 10. Success probability analysis.

the end of the time slot, the system state is regarded as failure and waits for the arrival of RTCM message from any source. After the system receives the RTCM message at the failure state, it converts to the success state and resumes the time slot timer. There are some discussions in the parallel system as follows.

- **Time slot size:** Considering other RTK-GPS based systems, at the viewpoint of both base station and the receiver, it is better to set T as 1 second, and yield the RTCM message propagation as 1 Hz frequency. Even though our proposed system can theoretically scale down T to $1/n$ when multiplexing n RTCM sources since each source tries to send RTCM in 1 Hz, it is recommended to receive at 1 Hz [15]. This is because the RTCM reception over 1 Hz could be redundant or harmful for the error compensation of the GPS result.
- **RTCM message selection:** If multiple RTCM messages arrive in the same slot, the system should select which one is propagated to the network. We designed a simple priority rule for RTCM selection, exploiting the observation that each RTCM message source affects the error correction performance of the RTK-GPS. At first, the **ground station (or GCS)** derives the cluster point of the position of the vehicles in the —RTCM message propagation network. Then, it selects the RTCM message source whose reference point is the nearest to the cluster point. In our propagation system, the position of the cluster point may change by the vehicles' movements. The parallel system can keep track of the RTCM message source which results in the best error correction.
- **RTCM message fragmentation:** The size of RTCM message data is variable. While running the network RTK, we observed that the RTCM message is often fragmented, so the delay between the fragments cannot be ignorable. In our implementation, we ignored the fragmented RTCM message over two time slots, since a part of the RTCM message cannot be interpreted at the vehicles.

To derive the availability of the system, we analyze the failure rate of the RTCM message reception. Success and failure condition of the system is graphically represented in Fig. 10. For simplicity, we start from the case of the single RTCM message source. We assume that the arrival time of the last RTCM message is uniformly distributed along a single time slot space $[0, T]$. Then, we let t be the time interval between the start point of a time slot including the last RTCM message and the arrival time of the message. If the next RTCM message arrives in $2T - t$, then the system maintains

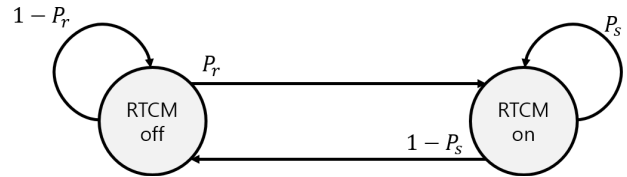


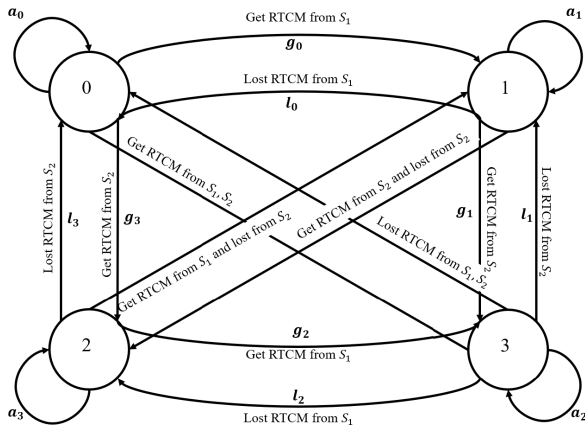
FIGURE 11. Markov chain of the parallel system with single RTCM message source.

the successful state at the next time slot. To sum up, we can derive the success probability of the system P_s as

$$P_s = \int_0^T P(t_i < (2T - t))dt \tag{9}$$

where t_i refers to the inter-delay between the consecutive RTCM messages. Also, if the arrival time of delayed RTCM message arrives at the failure state, the system changes to the successful state. We let P_r refer to the probability of the delayed RTCM message's arrival. As aforementioned, the probability of the arrival time can be approximated to the exponential distribution, so P_r can be derived to a constant value for each RTCM message source. Our Markov model is *discrete-time Markov Chain*, due to the existence of the time slot T . Graphical representation of the model with single RTCM message source is shown in Fig. 11. At Fig. 11, P_s is adopted to the transition probability for success-to-success rate, and success-to-failure rate. However, failure-to-success transition rate referred to P_r should be derived differently, since the recovery of the system occurs immediately when the next RTCM message is arrived. P_r can be regarded as the probability that the RTCM message arrives while the system is in failure state, which means $t_i > T$.

We plotted a PDF of $\{t_i | t_i > T\}$ using the measurements in Fig. 2, as Fig. 3 and fit the PDF to exponential distribution. As shown in Fig. 3, the distribution of t_i can be estimated to the exponential distribution, expressed to $f(x; \lambda) = \lambda e^{-\lambda x}$. Also, we can conclude that $P_r = 1/\lambda$, where λ can be derived from the fitting. By extending the single-source Markov model, Fig. 12 illustrates the Markov chain of the system with two RTCM message source. S_i refers to an activated RTCM message source, where $i \in N$. Note that the system can maintain the success state when at least one RTCM message arrives at each time slot. Transition probability matrix of the system \mathbf{P} can be expressed to Eq. (10), as shown at the bottom of the next page, where $P_{s,i}$ refers to the success probability and $P_{r,i}$ refers to the failure recovery probability of S_i . From Eq. (10), we can derive the steady-state probability vector π of the system by Eq. (11), as shown at the bottom of the next page. Fig. 13 shows the availability of the parallel systems with one or two RTCM message sources. In the figure, we assumed that P_r and P_s are the same along the RTCM message sources. From the Markov chains modeled in Figs. 11 and 12, we calculated the steady-state probabilities with respect to the P_r, P_s using Eqs. (10), (11), and summed the probabilities that represent the arrival of at least one of the RTCM messages. The figure shows the improvements of



State 0 : No RTCM
 State 1 : RTCM only from S_1
 State 2 : RTCM only from S_2
 State 3 : RTCM from S_1, S_2

FIGURE 12. Markov chain of the parallel system with two RTCM message sources.

the availability when two RTCM message sources are used, with the same performance among them. It means, although the case of one RTCM message source can provide high availability when P_r and P_s are large, the case of two RTCM message sources amplifies the availability with the same environmental factors. The case studies of using one or multiple practical RTCM message sources having the different factors is addressed in Section V.

IV. IMPLEMENTATION

In order to implement our proposed system, we use the off-the-shelf equipments listed in Table 2. Using these modules, we configured the RTK Base station and UAV as shown in Fig. 14. When constructing the local base station shown in Fig. 14a, we implemented the RTCM message sources in two types. The first method directly connected the RTK base station module to the GCS. The second method connected the RTK base station module and the network interface card to the Odroid XU4. In this case, the network interface card (NIC) was required for the network connectivity with the GCS. When constructing UAV shown in Fig. 14b, we used the DJI F550 platform and was equipped with a Pixhawk 2 as a flight controller and a Here+ RTK-GPS Rover module as GPS. In addition, we used Panda wireless PAU 07 as an NIC to configure the adhoc network between UAVs and GCS, and we mounted Odroid XU4 to use our software. We also produced and experimented with the UGV to show that the system is available not only in UAVs but also in other mobile devices.

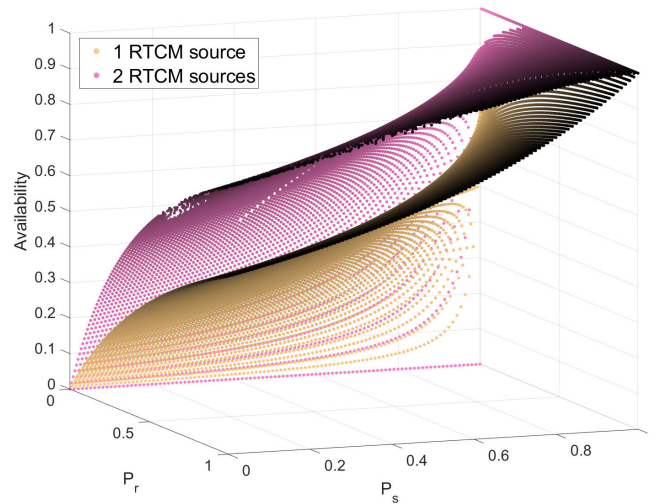


FIGURE 13. Availability comparison of the RTCM multiplexing.

TABLE 2. The specification of the prototype.

	Component	Model
Local base station	GPS	Here+ RTK-GPS base
	Companion board	ODROID XU4
UAV	GPS	Here+ RTK-GPS rover
	Frame	DJI F550
	Controller	Pixhawk 2
	NIC	Panda wireless - PAU 07
UGV	Companion board	ODROID XU4
	GPS	Here+ RTK-GPS rover
	Frame	Turtlebot3 - burger
	Controller	OpenCR
GCS		Pixhawk 2
	NIC	Panda wireless - PAU 07
	Laptop	LG-gram
	NIC	Panda wireless - PAU 07

A. RTCM PROPAGATION

We developed the system that receives the RTCM messages from the RTCM sources (local base station or NTRIP) and forwards through the vehicle network.

- 1) The GCS sent raw RTCM messages to the vehicles, and the vehicles input the message in the ROS topic defined in MAVROS, which forwards RTCM message to the Pixhawk 2 by MAVLink protocol.
- 2) The NTRIP Client was needed to connect to the actual NTRIP server, so we utilized the open source code published in github [16].
- 3) The method using the local base station had to implement both the server and the client. To implement

$$\begin{bmatrix} (1 - P_{r,1})(1 - P_{r,2}) & P_{r,1}(1 - P_{r,2}) & (1 - P_{r,1})P_{r,2} & P_{r,1}P_{r,2} \\ (1 - P_{s,1})(1 - P_{r,2}) & P_{s,1}(1 - P_{r,2}) & (1 - P_{s,1})P_{r,2} & P_{s,1}P_{r,2} \\ (1 - P_{r,1})(1 - P_{s,2}) & P_{r,1}(1 - P_{s,2}) & (1 - P_{r,1})P_{s,2} & P_{r,1}P_{s,2} \\ (1 - P_{s,1})(1 - P_{s,2}) & P_{s,1}(1 - P_{s,2}) & (1 - P_{s,1})P_{s,2} & P_{s,1}P_{s,2} \end{bmatrix} \tag{10}$$

$$\pi \mathbf{P} = \pi. \tag{11}$$

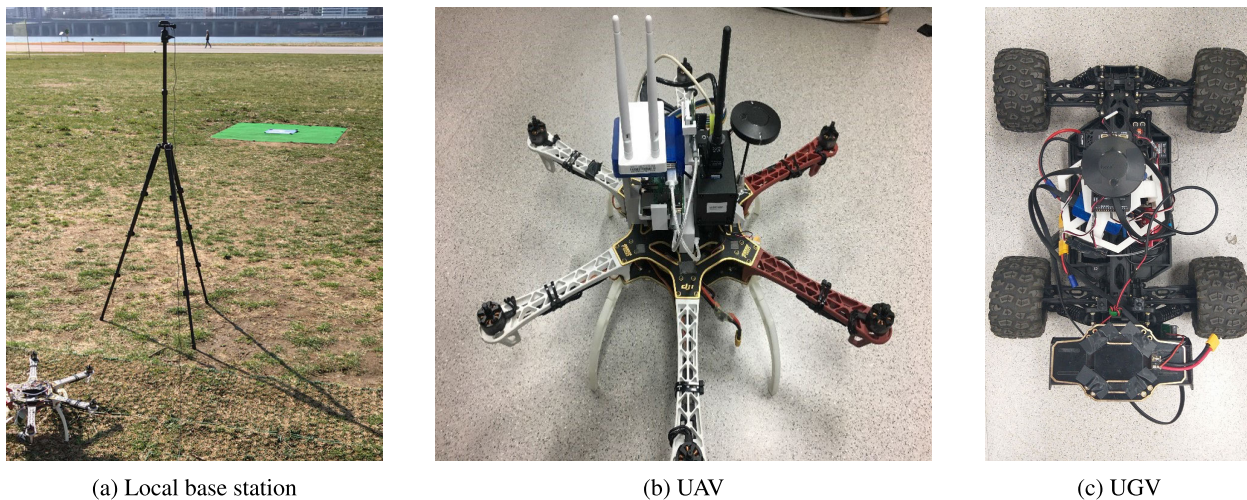


FIGURE 14. The prototypes for experiments.

the server, we used the RTKLIB [17] which is also open source. We used the version 2.4.3 b31, because it supports the ublox m8p chip which used at the Here+ RTK-GPS. We implemented a server that sends the string message received from Here+ RTK-GPS to the GCS with RTKLIB. Also, we created a client to receive the message at GCS.

- 4) We constructed a Wi-Fi ad-hoc network by attaching NICs to the GCS and the unmanned vehicles. The GCS propagated the data using this wireless ad hoc network connection. By doing so, a drone can receive the RTCM if it can establish a Wi-Fi link to at least one neighbor node, either drone or GCS. Note that we implemented a flying ad hoc network based with UAVs and a ground ad hoc network with UGVs for demonstration, and therefore the resulting network topology can be varied.

B. COLD-STANDBY IMPLEMENTATION

In the case of cold-standby (Section III-B.2), we implemented a network socket that initiates the connection when switching. If the master system becomes unstable or disconnected, then it switches to the connection with one of the slave systems and receive the RTCM messages. The pseudo code of the cold-standby is represented in Algorithm 1.

C. HOT-STANDBY IMPLEMENTATION

In the case of hot-standby (Section III-B.3), we pre-connected a socket to receive the RTCM messages from each NTRIP mount and the base station in GCS. By doing so, the data streams were continuously received, but we should periodically check and clear the socket buffer of the slave nodes. In other words, we implemented the system that the data from the slave node was ready to be transferred but neither used nor accumulated. If the system is implemented in this state, when we execute the system if an RTCM source *A* becomes unstable or disconnected while the GCS received data from *A*

Algorithm 1 Cold-Standby Switching Algorithm

```

1: Function RTCM message receiver
2: while exception not occurred do
3:   if already connected to a RTCM message source then
4:     wait
5:   else
6:     break
7:   end if
8: end while
9: try:
10:  connection open and RTCM source load
11: except:
12:  RTCM source load fail
13: while exception not occurred do
14:  try:
15:    receive RTCM message
16:  except connection timeout or another error:
17:    switching another RTCM message source
18:  else:
19:    inject RTCM message to vehicle via MAVLink
20:    message(in mavros)
21:  finally:
22:    connection close
23: end while

```

through a socket connection, it switches to the other RTCM source *B* transfers the RTCM messages. The pseudo code of the hot-standby is represented in Algorithm 2.

D. PARALLEL SYSTEM IMPLEMENTATION

In the case of the parallel system (Section III-B.5), GCS received all RTCM messages from the connected sockets, selected the RTCM message from them, and propagated to the vehicles through the network. If there was no RTCM message received for time *T*, it wait for the message reception

Algorithm 2 Hot-Standby Switching Algorithm

```

1: Function RTCM message receiver
2:   try:
3:     connection open and RTCM message source load
4:   except:
5:     RTCM message source fail
6:   while exception not occurred do
7:     if already connected to a RTCM message source then
8:       receive RTCM messages for a moment and empty
9:       the buffer
10:    else
11:      break
12:    end if
13:  end while
14:  while exception not occurred do
15:    try:
16:      receive RTCM message
17:    except connection timeout or another error:
18:      switching another RTCM message source
19:    else:
20:      inject RTCM message to vehicle via MAVLink
21:      message(in mavros)
22:    finally:
23:      connection close
24:  end while

```

and sends the first RTCM to the vehicles. The pseudo code of the parallel system is represented in Algorithm 3. Note that parallel system does not yield the system failure detection time but *repair time*, which is the time to receive the delayed RTCM message (line 26). On our implementation, we selected a latest RTCM message in line 28.

V. EVALUATION

We numerically evaluated the proposed redundancy systems discussed in Section III. We analyzed the experimental results of the switching systems and the parallel system, to show the validity of our proposal. We analyzed the results regarding three metrics: availability, robustness and the position accuracy.

A. RTCM AVAILABILITY**1) SWITCHING SYSTEM WITH HOT-STANDBY AND COLD-STANDBY COMPONENTS**

In Fig. 3, we showed that the failure-to-success transition probability can be approximated to the exponential distribution. Using this property, we obtained the failure rate and the MTTF as shown in Table 3. The table shows that the local base station definitely results in the lowest failure rate, while the NTRIP servers located at the island (i.e., DOKD and JEJU) results in the higher failure rates. Although not as much as the island, the NTRIP sources in the urban areas also have relatively high failure rate. This result indicates that the RTCM message reception is significantly affected by the

Algorithm 3 RTCM Parallel Algorithm

```

1: Function RTCM message receiver
2:   connection open
3:   while exception not occurred do
4:     try:
5:       connection accept
6:     except:
7:       continue
8:   end while
9:   try:
10:    while exception not occurred do
11:      try:
12:        receive and store RTCM message
13:      except:
14:        try reconnect
15:        inject RTCM message to vehicle via MAVLink
16:        message(in mavros)
17:      end while
18:    finally:
19:      connection close
20:  Function RTCM parallel loop
21:  while exception not occurred do
22:    Wait for the time  $T$ 
23:    Collect newly received RTCM messages
24:    if no RTCM message received then
25:      wait for RTCM message reception
26:    end if
27:    Select one of the RTCM messages
28:    inject RTCM message to vehicle via MAVLink
29:    message(in mavros)
30:  end while

```

TABLE 3. The failure rate & MTTF according to Mount.

Mount	Failure rate(%)	MTTF(sec)
Local base station	0.594457	169.902954
NTRIP_SOUL	4.036995	25.018609
NTRIP_SUWN	2.124734	47.535362
NTRIP_HONC	3.774528	26.758313
NRTIP_DOKD	35.320585	2.859522
NTRIP_JEJU	17.835270	5.662936

method and the status of the connection. From this reason, we confirmed that the RTK-GPS needs a redundancy system for robustness.

Because RTCM messages were supposed to be sent at 1 Hz, we set the timeout deadline of the RTCM arrival latency to 1.01 seconds with a slight margin. That is, in Eq. (6), the $\tau = 1.01$. Then, we obtain the availability of cold-standby and hot-standby using Table 3 and Eqs. (6-8). The values appear as Table 4. Comparing with Table 3, we can observe that the availability of the switching systems outperforms the case of the system using single source. In particular, the cases of the hot-standby have been notably improved. Based on these results, we confirmed that the proposed switching system

TABLE 4. The availability of cold-standby and hot-standby according to the mounts.

Source1 & Source2	Availability	
	Cold-standby	Hot-standby
Local base station & NTRIP_SOUL	98.917056	99.809210
Local base station & NTRIP_SUWN	99.028111	99.828934
Local base station & NTRIP_HONC	98.926534	99.810895
Local base station & NRTIP_DOKD	98.779850	99.784792
Local base station & NTRIP_JEJU	98.799099	99.788221
NTRIP_SOUL & NTRIP_SUWN	97.142779	99.489075
NTRIP_SOUL & NTRIP_HONC	96.041618	99.285516
NTRIP_SOUL & NRTIP_DOKD	92.889542	98.681096
NTRIP_SOUL & NTRIP_JEJU	93.496984	98.800160
NTRIP_SUWN & NTRIP_HONC	97.207818	99.500980
NTRIP_SUWN & NRTIP_DOKD	95.937473	99.266066
NTRIP_SUWN & NTRIP_JEJU	96.143301	99.304473
NTRIP_HONC & NRTIP_DOKD	93.279129	98.757604
NTRIP_HONC & NTRIP_JEJU	93.824383	98.863811
NRTIP_DOKD & NTRIP_JEJU	79.974584	95.811157

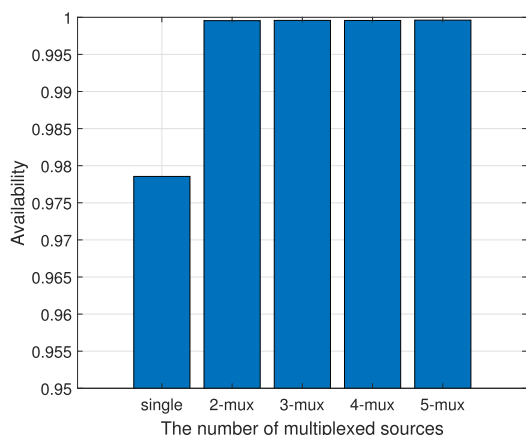


FIGURE 15. The availability probability according to the number of multiplexed sources.

can keep the RTCM message arrival time for the vehicles to maintain the accurate RTK-GPS navigation.

2) PARALLEL SYSTEM

In order to show the improvement of the availability in the case of the parallel system, we used five NTRIP servers which were used for the evaluation of the switching system. As shown in the Fig. 15, it could be seen that the availability was approximately 97.86% when only one RTCM message source was used, 99.95% when operating two RTCM message sources, and 99.98% when operating more than three RTCM message sources. From the observation, we could conclude that the more RTCM message sources used for the parallel system, the availability comes closer to 1. Furthermore, by using this to increase the number of RTCM message sources, the message propagation rate could also be adjusted. Note that we used multiple local base stations and NTRIP servers to demonstrate the effect of the parallel system, but this could greatly differ the correction data among the RTCM message sources. This problem can be resolved by using only NTRIP servers located in nearby areas, or the combination with local base stations and virtual base stations such as VRS. In addition, granting the priority of incoming RTCM messages according to distance between the drones and the



FIGURE 16. The actual flight of the UAVs.

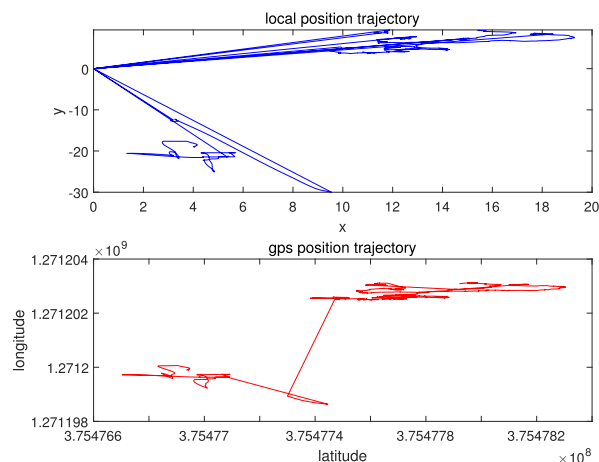


FIGURE 17. GPS loss during the switching experiment due to the burst of the RTCM messages.

sources can be the proper approach to prevent the problem. In conclusion, the results showed that our proposed switching system and parallel system improve the availability of RTK-GPS.

B. RTCM ROBUSTNESS

We showed the robustness by demonstrating our system in practical environments (14,16, 18-20). We implemented our proposed design (Fig. 1) using the prototypes that we made. Then, we conducted the experiment using them like Fig. 16. We performed the experiment with our implementation, as shown in Fig. 16. In order to invoke the failure of the RTCM source, we detached the USB connection between the GCS and the local base station, which will cause the switching to the other source, NTRIP.

In the case of the hot-standby, the network socket receiving RTCM messages should be flushed periodically when it is in inactivated. Otherwise, when switching occurs, the stacked RTCM messages are burstly broadcast and paralyze the network or the RTK-GPS module. Fig. 17 shows the GPS loss of the vehicle due to the RTCM burst. In the figure, we observed that local position (the position value not fused with GPS) trajectory and GPS position trajectory do not coincide with



FIGURE 18. Detaching local base station during UAV flight and switching to NTRIP source.

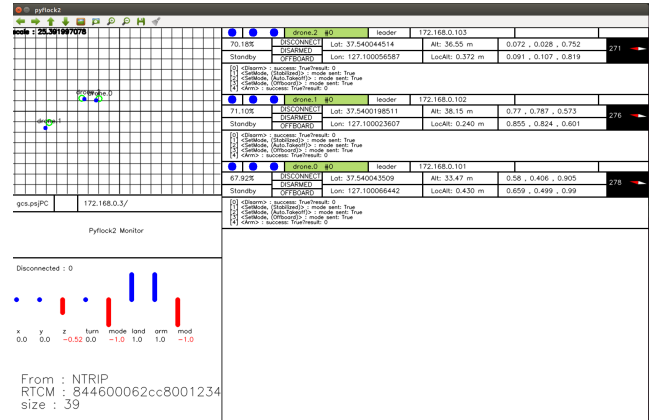


FIGURE 20. The screen when using the NTRIP Source.

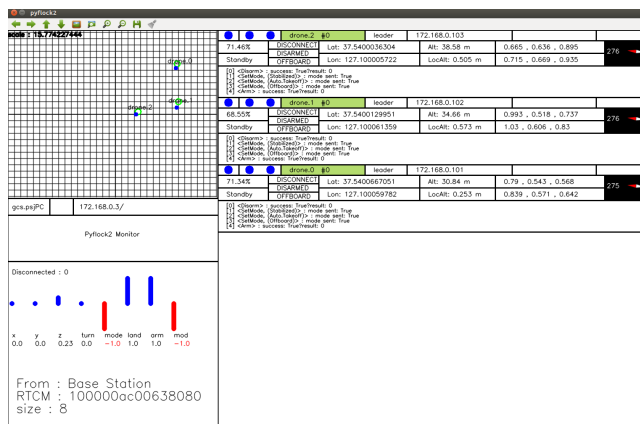


FIGURE 19. The screen when using the local base station.

each other. If we look more closely, the bottom left and the top right sections of the both trajectory are matched. This is because the RTCM messages were received excessively, so the local position value was momentarily lost, resulting in repeated zero values. We identified this symptom and prevented these problems through real world experiments.

As shown in Figs. 19 and 20, the switching from local base station to NTRIP was performed successfully. This is clearly shown by the trajectory comparison in Fig. 21, which shows that the GPS position trajectory and the local position trajectory are similarly drawn, even though RTCM message source switching was performed during flight.

C. LOCATION ACCURACY

Finally, after verifying and improving the availability and the robustness, we tested the performance of the RTK-GPS switching system. This experiment was designed to test the location accuracy and check if the RTCM message source switched with expected latency. We ran the experiment in two ways. The first experiment was carried out with a drone in a stationary state, and the second experiment used a UGV in a moving state.

Fig. 22 is the result of the stationary state experiment. In the Fig. 22, the RTCM source index indicates the identity

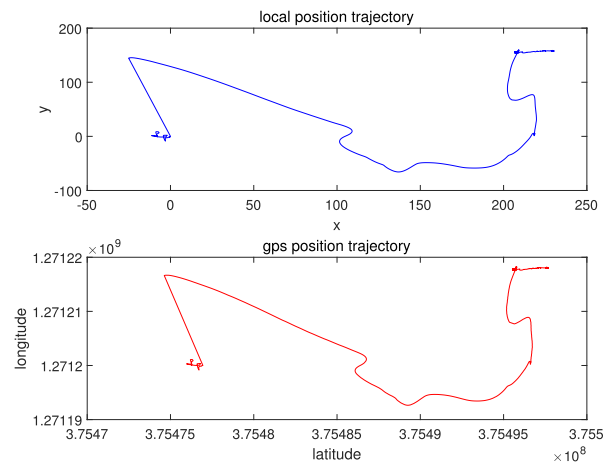


FIGURE 21. RTK-GPS maintenance due to the robust propagation of the RTCM cycle control.

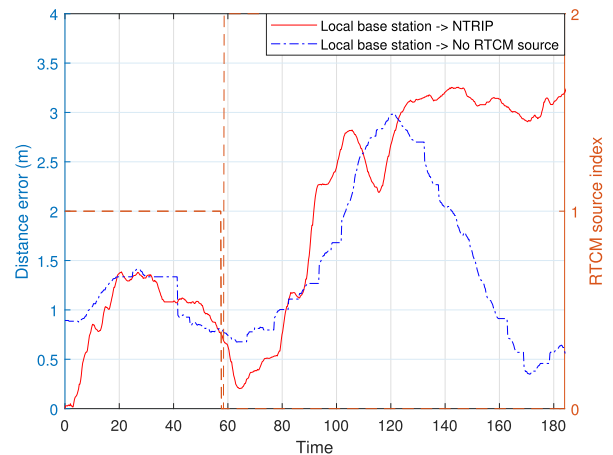


FIGURE 22. The distance error according to the RTCM source switching's success and fail. The RTCM source index indicates the identity of the RTCM source, 0 indicates the absence of the RTCM source, 1 indicates the local base station, and 2 indicates NTRIP.

of the RTCM source, where 0 indicates the absence of the RTCM source (GPS only), 1 indicates the local base station, and 2 indicates NTRIP. When the RTCM source index is 1,



FIGURE 23. The moving path of the UGV in the second experiment.

i.e. when the RTCM source is the local base station, we can see that the distance error in both cases are similar to 1m. However, when the RTCM source index is 0, i.e. when the RTCM source is not present, we can see that the distance error is *fluctuating* as shown in the blue line. On the other hand, when the RTCM source index is 2, i.e. when the RTCM source is NTRIP, it can be seen that the distance error *converges* to 2m. This result is because the correction data received from the local base station and the NTRIP Client differs by about 1m, and the drone’s location was adjusted for the correction data after the RTCM source had changed. In other words, checking the graph after the red line has stabilized, it is possible to confirm that the error comes to be within 1 m when RTCM messages are received from the NTRIP Client. Through this experiment, we verified that the switching module makes the GPS value more stabilize than the GPS-only case, although there was a little bit of position shift due to the different correction data. Also, we showed that our switching system implementation works seamlessly.

Prior to the second experiment, we decided on the experiment path as shown in Fig. 23 to compare the trajectory. While moving the UGV following the path shown in the figure twice, we collected the RTK-GPS (or GPS-only) measurements. We conducted the experiment following two cases of scenarios. In the first case, the GCS switches to the NTRIP at the local RTCM failure after the first round, and the UGV maintains the RTK-GPS. In the second case, the GCS loses the local RTCM message by the cable disconnection after the first round, and the UGV switches to normal GPS state.

Fig. 24 compares the trajectory of each scenario, where the distance unit is converted to meters. As shown in the figure, RTCM switching case (the first case) maintains the accurate positioning of the UGV, while the RTCM failure case (the second case) shows relatively large fluctuation at the trajectory. Although the RTCM switching case shows some large error when switching (right-bottom edge of the rectangular course), the UGV stabilizes the position through the NTRIP. We calculated the average error of each trajectory, which results in 44.04cm at the first case and the 62.69cm

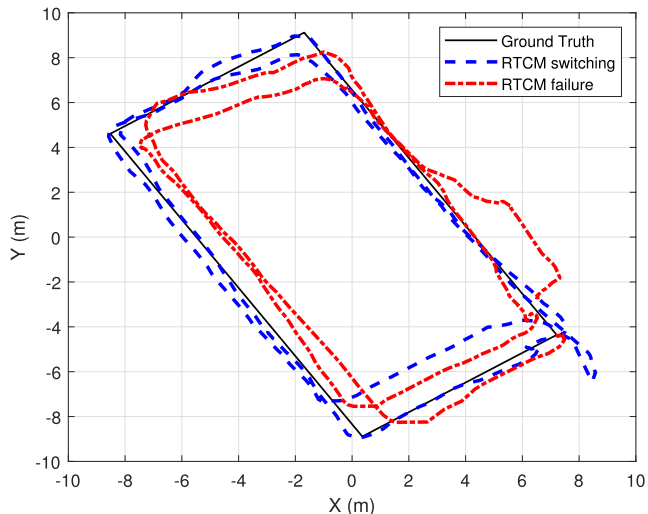


FIGURE 24. Trajectory comparison of the second experiment.

at the second case. This figure indicates that our proposed system reduces about 30% of the error by the RTCM redundancy. This experiment showed that the proposed system maintains RTCM message reception by redundancy even if RTCM source failure occurs, thereby increasing the availability of RTK-GPS and improving the accuracy of multi-drone navigation.

D. DISCUSSIONS

Based on the analysis and experiments on the proposed system, it is necessary to discuss its main features and feasibility in various usage scenarios.

- The position loss shown in Fig. 17 may be an unexpected situation when developing the device, and this may not happen on other devices. In our experiments, this GPS loss can lead to unpredictable drone flight, which can lead to serious accidents in practical drone operation. To avoid this situation, the drone-level RTCM message reception management module should be more sustainable in various cases.
- According to the availability analysis in Fig. 15, a two-source parallel system produces almost the same results as three or more sources. However, if the candidate RTCM sources’ failure rates are low, due to the harsh network environment, the two-source parallel system could not guarantee the required availability to the drones. The proposed analysis of the parallel system approach can present the availability of n-source cases, which allows us to flexibly design parallel systems in various network environments.
- Developed through redundancy, the proposed system can integrate received RTCM messages and propagate more useful correction information to drones. In addition to source-to-source integration, time-varying smoothing of the correction data can prevent the large shift of the drone position during the switching process, observed in Figs. 22 and 24. Our future study focuses on

RTCM message integration and refinement of correction data to improve the accuracy of multi-drone positioning.

VI. CONCLUSION

We proposed a novel design of RTCM propagation network for the multi-drone environments, including the redundancy system which can improve the availability while using RTK-GPS. First, we clarified that it is possible to propagate the RTCM messages to multiple drones through the wireless mesh network, and switch the RTCM message sources during the propagation. Second, we found that when cold-standby or hot-standby was employed, the switching time was short enough to maintain the RTK-GPS functions. Finally, we have shown that switching or parallel systems can be used to ensure the availability of RTCM message reception and can also speed up RTCM message propagation.

Our proposed system uses local base stations and NTRIP servers. Like the traditional RTK-GPS, this structure limits mobility and usability. Thus, we will continue the studies to improve the performance of RTK-GPS by using one of the multi-drone that have moved to the operation area as a base station. We will set up an RTK-GPS base station on UAV and attempt to use it as a mobile RTCM message source. In other words, it aims to be able to switch the vehicle to the base station in the landing state and the mobile vehicle in the flight state. This allows us to use RTK-GPS even if the installed RTCM message source is not nearby. We hope that the proposed RTCM propagation system design will inspire the development of navigation research around the world.

REFERENCES

- [1] T. L. Dammalage, P. Srinuandee, L. Samarakoon, J. Susaki, and T. Srisahakit, "Potential accuracy and practical benefits of NTRIP protocol over conventional RTK and DGPS observation methods," *Map Asia*, vol. 29, Aug. 2006. [Online]. Available: <https://www.geospatialworld.net/article/potential-accuracy-and-practical-benefits-of-ntrip-protocol-over-conventional-rtk-and-dgps-observation-methods/>
- [2] U. Vollath, H. Landau, X. Chen, K. Doucet, and C. Pagels, "Network RTK versus single base RTK—understanding the error characteristics," in *Proc. 15th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Portland, OR, USA, 2002, pp. 24–27.
- [3] C. Rizos, "Network RTK research and implementation: A geodetic perspective," *J. Global Positioning Syst.*, vol. 1, no. 2, pp. 144–150, 2002.
- [4] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*. Springer, 2012.
- [5] Y. Gao and A. Wojciechowski, "High precision kinematic positioning using single dual-frequency GPS receiver," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 34, pp. 845–850, Jul. 2004.
- [6] C. L. Hay, "Method and system for global positioning system mask angle optimization," U.S. Patent 6 580 390, Jun. 17, 2003.
- [7] *RTCM Standard 10403.3 Differential GNSS (Global Navigation Satellite Systems) Services-Version 3*, document RTCM Special Committee 104, 2016.
- [8] G. Weber, D. Dettmering, and H. Gebhard, "Networked transport of RTCM via Internet protocol (NTRIP)," in *A Window on the Future of Geodesy*. Berlin, Germany: Springer, 2005, pp. 60–64.
- [9] A. El-Mowafy, "Precise real-time positioning using Network RTK," in *Global Navigation Satellite Systems: Signal, Theory Application*. Rijeka, Croatia: InTech, 2012.
- [10] N. Brown, R. Keenan, B. Richter, and L. Troyer, "Advances in ambiguity resolution for RTK applications using the new RTCM V3. 0 Master-Auxiliary messages," in *Proc. ION GNSS*, vol. 13, 2005, p. 16.
- [11] M. Bakuła, R. Pelc-Mieczkowska, and M. Walawski, "Reliable and redundant RTK positioning for applications in hard observational conditions," *Artif. Satell.*, vol. 47, no. 1, pp. 23–33, Jan. 2012.
- [12] T. Speth, A. Kamann, T. Brandmeier, and U. Jumar, "Precise relative ego-positioning by stand-alone RTK-GPS," in *Proc. 13th Workshop Positioning, Navigat. Commun. (WPNC)*, Oct. 2016, pp. 1–6.
- [13] P. Large, G. R. Kirk, and M. T. Allison, "Method and system for variable data rate transmission in RTK GPS survey system," U.S. Patent 6 985 104, Jan. 10, 2006.
- [14] A. Y. Chung, J. Jung, K. Kim, H. K. Lee, J. Lee, S. K. Lee, S. Yoo, and H. Kim, "Poster: Swarming drones can connect you to the network," in *Proc. 13th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2015, p. 477.
- [15] U.-B. Ag. (2017). *u-blox M8 High Precis. GNSS Modules Data Sheet*. [Online]. Available: https://www.u-blox.com/sites/default/files/NEO-M8P_DataSheet_%28UBX-15016656%29.pdf
- [16] jcmb/ntrip. [Github] Retrieved From. Accessed: Apr. 24, 2020. [Online]. Available: <https://github.com/jcmb/NTRIP/blob/master/NTRIP%20Client/NtripClient.py>
- [17] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *Proc. Int. Symp. GPS/GNSS*, Jeju, South Korea, 2009, pp. 4–6.



INSEOP UM was born in August 1992. He received the B.S.E. degree in electrical engineering from Kyunghee University, South Korea, in 2018. He is currently pursuing the M.S.E. degree with the School of Electrical Engineering, Korea University, Seoul, South Korea. His research interests are in the areas of community wireless networks, drone localization, GNSS, and reinforcement learning for drone.



SEONGJOON PARK was born in September 1990. He received the B.S.E. degree in electrical engineering from Korea University, Seoul, South Korea, in 2015, where he is currently pursuing the Ph.D. degree with the School of Electrical Engineering. His research interests are in the areas of community wireless networks, network modeling and simulations, and multiple UAVs applications.



HYEONG TAE KIM was born in April 1997. He received the B.S.E. degree in electrical engineering from Korea University, Seoul, South Korea, in 2019, where he is currently pursuing the M.S.E. degree with the School of Electrical Engineering. His research interests are in the areas of swarm intelligence, deep neural networks particularly in reinforcement learning, and multiagent are inforcement learning.



HWANGNAM KIM received the B.S.E. degree in computer engineering from Pusan National University, Busan, South Korea, in 1992, the M.S.E. degree in computer engineering from Seoul National University, Seoul, South Korea, in 1994, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, in 2004. He is currently a Professor with the School of Electrical Engineering, Korea University, Seoul. His research interests are in the areas

of wireless networks, unmanned aerial systems (UAS), UAS traffic management (UTM), counter UAS systems, the Internet of Things, and cyber physical systems.

• • •