

論文

J. of The Korean Society for Aeronautical and Space Sciences 45(7), 610-617(2017)

DOI:https://doi.org/10.5139/JKSAS.2017.45.7.610

ISSN 1225-1348(print), 2287-6871(online)

스마트폰을 활용한 소형 저가 유도탄 유도조종장치용 어플리케이션 개발

노정훈*, 조경국**, 김성준**, 김원섭**, 정진섭**, 상진우**, 박청운**, 공민식**

Development of application for guidance and controller unit for low cost and small UAV missile based on smartphone

Junghoon Noh*, Kyongkuk Cho**, Seongjun Kim**, Wonsop Kim**, Jinseob Jeong**,
Jinwoo Sang**, Chung-Woon Park** and Minsik Gong**

Agency of defense development* ,**

ABSTRACT

In the recent weapon system trend, it is required to develop small and low cost guidance missile to track and strike the enemy target effectively. Controlling the such small drone typed weapon demands a integrated electronic device that equipped with not only a wireless network interface, a high resolution camera, various sensors for target tracking, and position and attitude control but also a high performance processor that integrates and processes those sensor outputs in real-time. In this paper, we propose the android smartphone as a solution for that and implement the guidance and control application of the missile. Furthermore, the performance of the implemented guidance and control application is analyzed through the simulation.

초 록

최근의 무기체계에서는 효율적으로 적진을 탐색, 타격할 수 있는 저가의 소형 유도무기 개발이 많이 요구되고 있다. 이처럼 저가의 소형 드론형 무기의 제어를 위해서는 무선 인터페이스, 자세제어 위치탐색 타깃 탐색들을 위한 다양한 센서들과 고성능 카메라, 그리고 이를 실시간으로 통합하여 처리할 수 있는 고성능의 프로세싱 능력들을 갖춘 통합 전자장치가 필수적이다. 본 논문에서는 이를 위한 솔루션으로 안드로이드 스마트폰을 통합 전자 제어장치로 제안하고, 유도조종 알고리즘을 수행할 어플리케이션을 구현한다. 제안된 어플리케이션을 실제 스마트폰에 올려 비행모의 시뮬레이션을 통해 성능을 분석한다.

Key Words : UAV(무인기), Drone(드론), Android, Smart phone(스마트폰), Guidance rocket(유도로켓), Real time android(실시간 안드로이드)

1. 서 론

최근의 무기체계는 정확한 유도성능을 유지하

면서 저가의 소형 유도무기 개발이 많이 요구되고 있다. 최근에는 다량의 소형 드론형 무기로 이루어진 LOCUST(LOW-Cost Unmanned aerial

† Received : May 4, 2017 Revised : June 27, 2017 Accepted : June 28, 2017

* Corresponding author, E-mail : jung.hoon.noh@gmail.com

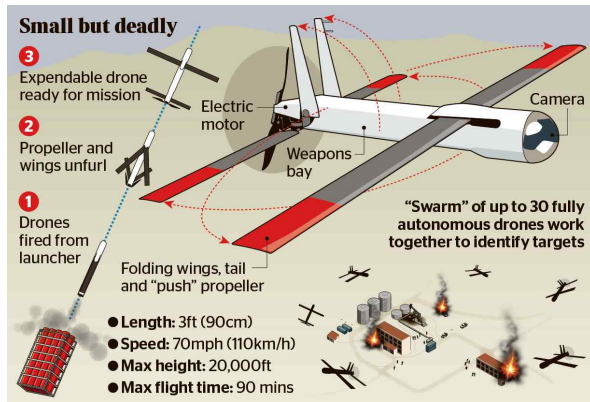


Fig. 1. US Navy LOUST Concept

vehicle Swarming Technology)가 미 해군에서 개발되기도 하였다[1]. Fig. 1과 같이 30대 이상의 저가의 드론들이 작전지역 상공을 비행하며, 목표 타격을 추적, 인지, 타격을 협력하여 진행한다. MBDA사에서는 2030년을 목표로 하는 미래 무기 컨셉 비전 프로젝트로 소형 UAV Missile, VIGILUS를 발표하기도 하였다[2]. VIGILUS는 다수의 드론이 단독 혹은 협력으로 타격을 탐색, 타격하는 시스템이다. 또한 미 국방부에서는 최근 F/A 18 슈퍼호넷에서 방출한 103대의 초소형 드론 프로젝트 퍼디스(Perdix)의 자율편대비행 실험에 성공했다고 밝히고 관련 영상을 공개한 바 있다[3]. 군사 전문가들은 이러한 드론은 저렴한 비용으로 제조할 수 있어 대량으로 비행시킬 경우 적대세력의 효율적으로 방위력을 제압할 수 있을 것이라는 기대를 하고 있다.

이와 같은 저가의 소형 드론형 무기의 제어를 위해서는 무선 인터페이스, 자세제어, 위치탐색, 타격 탐색들을 위한 다양한 센서들과 고성능 카메라, 그리고 이를 통합하여 처리할 수 있는 고성능의 프로세싱 능력 등을 갖춘 통합 유도조종장치가 필수적이다. 이를 위한 유도조종장치의 개발은 쉽지 않다. 다양한 기능들을 모두 통합할 수 있으면서도 소형화와 저가화를 달성할 수 있는 하드웨어와 소프트웨어의 개발은 오랜 시간, 노력, 인력과 물적 자원을 필요로 한다. 또한 무기체계의 특성상 안정성이 필수로 보장되어야 하는데, 이를 위한 검증과 최종적으로 안정성이 확보되는 데까지 들이는 노력과 시간은 이를 더 어렵게 한다.

본 논문에서는 이를 위한 솔루션으로 안드로이드 스마트폰을 통합 유도조종장치로 제안한다. 최근 몇 년 사이에 아이폰을 비롯한 여러 안드로이드 기반의 스마트폰은 엄청난 속도로 우리의 생활 깊은 곳에 자리 잡았다. 이러한 스마트폰은

Table 1. Nexus 5x hardware specification

Processor	Qualcomm® Snap dragon™ 808 processor, 1.8GHz Hexacore 64bit, Adreno 418 GPU
RAM	2GB LPDDR3
Flash	16GB, 32GB
Wireless interface	LTE cat. 6 Wi-Fi 802.11 a/b/g/n/ac 2x2 MIMO, Dual band (2.4GHz, 5.0GHz) Bluetooth 4.2 NFC
Location	GPS / GLONASS Digital compass
Sensor	finger print sensor Android sensor hub Accelerometer Gyroscope Barometer Proximity sensor Ambient light sensor Hall sensor

소형이면서 고성능 프로세서, LTE, Wi-Fi, Bluetooth 등의 무선인터페이스, 가속도, 자이로스코프, 기압계, 근접 센서, 광센서 등의 다양한 센서 탑재로 제안되는 무기체계에 요구되는 성능을 충분히 만족시켜준다. 대량생산에 의한 저가화와 상업용으로 이미 검증된 하드웨어라는 사실은 이를 더 돋보이게 한다.

이미 많은 부분에서 스마트폰을 제어장치 또는 조종 장치로 이용하려는 시도가 있어왔다. 드론이나 로봇의 원격조종장치로의 이용[4-8]은 말할 것도 없이, 드론이나 UAV에 탑재되어 비행제어, 자세제어 장치로 많은 연구가 이루어지고 있다[9-12]. [13]에서는 무인기의 영상추적 시스템으로 스마트폰 이용을 제안하기도 하였다. 심지어는 스마트폰을 이용한 인공위성이 개발된 사례도 있다[14-16]. 이런 사례들을 통해 스마트폰을 탑재한 유도무기체계의 실현가능성을 엿볼 수 있다.

본 논문에서는 스마트폰을 유도조종장치로 사용시 이에 대한 시스템 구성과 유도조종항법 어플리케이션 구성을 제안한다. 제안된 어플리케이션을 실제 스마트폰에 올려 비행모의 시뮬레이션을 통해 성능을 분석한다. 실제 유도무기 체계에 쓰이는 임베디드 장치에는 실시간 처리가 필수적으로 요구된다. 기본적인 안드로이드는 실시간 기반의 커널이 아닐뿐더러 자바 가상머신에서 실행

시간을 지원하지 않아, 이에 대한 한계점이 분명히 존재한다. 이를 극복하기 위해, 실시간 기능이 추가된 RT-Android를 적용하여 성능을 분석한다 [17, 18].

II. 시스템 구성

2.1 하드웨어 구성

본 논문에서 시뮬레이션을 위해 구글의 레퍼런스폰인 Nexus 5x를 사용한다. Table 1에서는 Nexus 5x의 하드웨어 스펙을 보여준다. Nexus 5x는 1.8GHz 쿼드코어 프로세서를 탑재하고, 가속도, 자이로, 기압계 등 비행의 자세제어에 필요한 기본적인 센서들을 갖추고 있다. 외부 장치와 연동하기 위한 입출력 인터페이스로는 유선은 USB, Audio, 무선으로는 Bluetooth, Wi-Fi, NFC 등이 있다. [11, 12]에서는 별도의 입출력 보드 (IOIO)를 USB를 통해 스마트폰에 연결하여 PWM(Pulse Width Modulation) 신호를 구동모터에 전달한다. 본 논문에서는, 유도조종항법 알고리즘의 스마트폰위에서의 구현과 성능검증이 주된 목적이므로 이러한 인터페이스들은 다루지 않는다.

Table 2에서는 Nexus 5x에 탑재된 MPU-6515의 가속도와 자이로 센서 성능을 보여준다. 모바일기에 장착될 목적으로 제작된 센서로 고속 동작을 필요로 하거나, 정밀한 항법성능을 필요로 하는 플랫폼에서는 사용되기 어렵다. 하지만 소형 UAV나 큰 가속도를 필요로 하지 않는 무기체계에는 충분히 적용가능하며, 유사한 성능의 센서 적용사례들도 어렵지 않게 찾아볼 수 있다. 일례로 Syntronics 사의 Digital Semi-Active Laser Seeker (DSALS)의 IMU 스펙 [27] 을 보면 유사한 성능의 센서가 사용되었음을 확인할 수 있다.

2.2 안드로이드 특성

효율적인 어플리케이션 구현을 위해 스마트폰

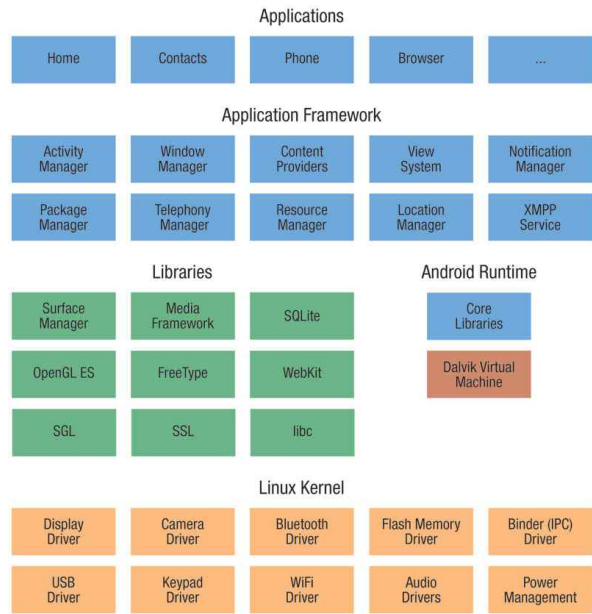


Fig. 2. Android structure

운영체제인 안드로이드의 구조와 특징에 대하여 살펴본다. 안드로이드의 전체적인 구조는 Fig. 2와 같다. 안드로이드의 핵심인 커널은 오픈소스 운영체제인 리눅스 기반이다. 리눅스 커널은 상당히 오랜 기간 개발, 관리되어 정교한 메모리 관리, 안정적인 멀티 스레드, 엄격한 보안 등의 고급 기능을 라이선스 없이 지원한다. 안드로이드의 리눅스 커널은 휴대폰에 맞게 개조되어, 크고 무거운 기능은 제거하고 알람, 디버거 등의 기능이 추가 되었다.

검증된 많은 오픈소스 라이브러리를 포함하고 있어 웬만한 기능은 별도의 외부 라이브러리를 사용할 필요가 없다. 자바가 기본적으로 제공하는 라이브러리 외에도 OpenGL, SQLite, FreeType 등의 서드 파티 라이브러리까지 모두 사용가능하다.

안드로이드는 자바 언어를 사용한다. 고수준 언어인 자바는 생산성이 높으며 하드웨어 추상층을 제공하여 하드웨어 전문 지식이 없어도 개발 가능하다. Android runtime은 리눅스 커널위에

Table 2. MPU-1615 specification

Gyro Full Scale Range (°/sec)	Gyro Sensitivity (LSB/°/sec)	Gyro Rate Noise (dps/Hz)	Accel Full Scale Range (g)	Accel Sensitivity LSB/g
±250	131	0.01	±2	16384
±500	65.5	0.01	±4	8192
±1000	32.8	0.01	±8	4096
±2000	16.4	0.01	±16	2048

서 자바 어플을 실행하기 위한 Dalvik virtual machine과 Core Library를 포함한다.

어플리케이션 프레임워크는 안드로이드 어플리케이션을 설치하고 실행하는데 기본적으로 필요한 프레임워크이다. 어플리케이션 구성의 기본 요소인 Activity, Content provider, View, Service 등을 관리하며, 사용자는 기본적으로 제공되는 이 프레임워크를 통해서 필요한 프로그램을 쉽게 제작할 수 있다.

안드로이드는 오픈소스 운영체제인 만큼 개발툴과 관련 문서가 모두 무료로 제공된다. 심지어 운영체제의 핵심 소스까지도 대부분 공개되어 있다. 오랜 기간에 걸쳐 검증된 리눅스 커널기반의 고급기능과, 검증된 많은 라이브러리 등을 라이선스 없이 사용할 수 있다는 점은 큰 장점이다.

하지만 한계 점도 분명히 있다. 안드로이드 플랫폼은 RTSJ(Real-Time Specification of Java) 표준을 준수하는 자바 가상머신을 제공하지 않고, 그 아래에서 실행하는 리눅스 커널은 PREEMPT_RT[19]와 같은 실시간 패치를 필수적으로 요구하지 않으므로, 실시간 어플리케이션을 작성하고 실행하는데 적합하지 않다. 따라서 안드로이드 플랫폼을 실시간 시스템에서 적용하고자 하면 타킷 어플리케이션의 실시간 요구사항을 충족시키기 어렵다.

안드로이드 플랫폼이 가진 장점을 활용하면서, 실시간 어플리케이션의 요구사항을 만족시키기 위해서 다양한 연구가 이루어졌다[17-22]. 본 논문에서는 실시간 안드로이드 적용에 따른 성능 분석을 위해, Real-Time android[17, 18]가 올려진 Raspberry-pi에서 개발된 어플리케이션의 실시간 성능을 비교한다.

III. 안드로이드 기반 유도조종항법 어플리케이션

3.1 시스템 디자인

스마트폰 기반의 유도조종 어플리케이션의 기능 및 성능을 확인하기 위하여 간단한 형태의 유도조종 알고리즘을 설계하고 미리 정해진 비행 시나리오에 대해 시뮬레이션을 진행한다. 이를 통해 유도조종 스마트폰의 실제 적용 가능성과 한계, 그리고 적용 가능한 플랫폼 등을 가늠해 본다.

유도조종 알고리즘이 수행되기 위해서는 발사 정보, 표적정보 및 각종 유도탄 비행정보가 필요하다. 또한 유도조종 알고리즘은 유도탄 위치, 자

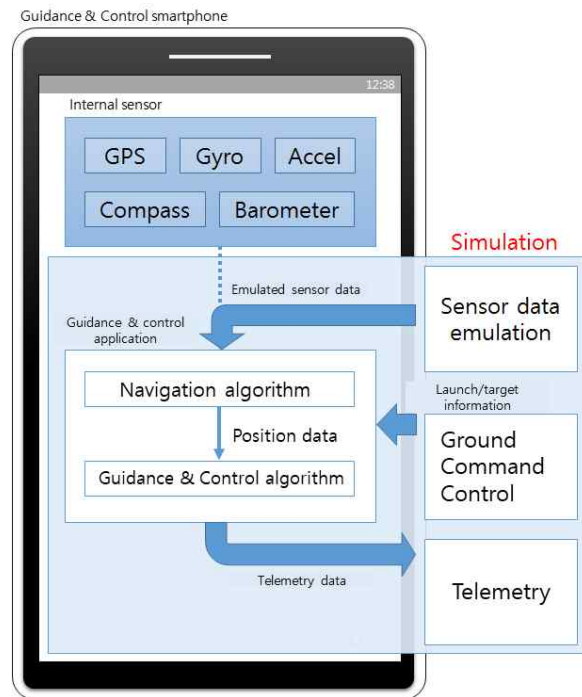


Fig. 3. System design

세, 속도 등의 정보를 요구하는데 이를 위해 항법 알고리즘을 어플리케이션에 탑재하였다. 이를 모의하기 위한 시스템 구성은 Fig. 3과 같다. 먼저 외부의 사격통제장치로부터 발사 절차에 따라 발사대 정보와 표적 정보를 입력 받으며 발사 이후에는 유도탄 내부의 센서정보(각속도, 가속도)를 일정 시간 간격으로 입력받아 일정한 빈도수 F로 알고리즘을 수행한다.

실제 비행에서는 스마트폰의 내부 센서 정보를 이용하지만, 시뮬레이션 상에서는 이를 외부의 유도탄 운동 모의 장치로부터 모의 센서데이터를 입력받도록 한다. 센서 데이터는 정해진 비행 시나리오에 대해 미리 수행된 시뮬레이션 데이터를 일정한 시간 간격 T로 전송하는 방법으로 모의하였다.

유도조종 알고리즘의 최종 출력인 구동 명령을 포함한 알고리즘의 각종 내부 신호, 비행정보는 외부의 원격측정장치로 전송하여 수행 결과를 확인할 수 있도록 구성하였다. 유도조종 스마트폰과 각각의 외부장치와의 인터페이스는 스마트폰의 WiFi를 통한 UDP 프로토콜을 이용한다.

3.2 어플리케이션 디자인

이번 장에서는 유도조종항법 어플리케이션 구조에 대해 설명한다. 전체적인 구조는 Fig. 4와 같다. 유도조종항법 어플리케이션은 크게 3개의 스레드 그룹으로 구성되어 있다.

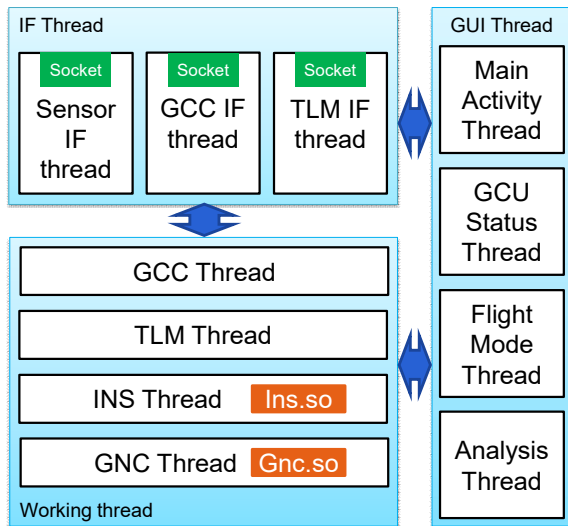


Fig. 4. Application structure

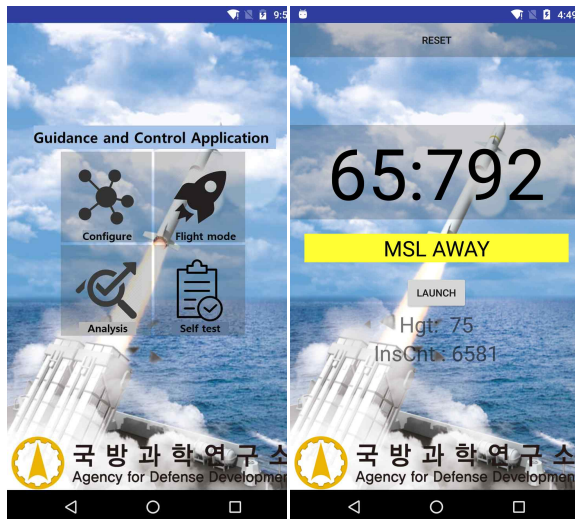


Fig. 5. Main activity Fig. 6. Flightmode activity

첫 번째로 touch pad를 통한 사용자와의 인터페이스를 구현한 GUI 스레드 그룹이다. 사용자는 Fig. 5, 6과 같은 GUI를 통하여 유도조종 스마트폰의 각종 설정들을 변경하거나, 유도탄의 현재 상태정보를 모니터링 할 수 있다. 두 번째는 외부장치들과의 인터페이스 스레드 그룹이다. 각각의 스레드는 각 외부장치와 유도조종스마트폰의 소켓통신을 담당한다. 세 번째로는 각각 독립된 작업을 수행하는 작업 스레드이다. 작업 스레드는 각각의 수행임무를 가지고 해당 작업에 대한 요청이 들어오면 해당 작업을 수행하고 결과를 리턴하거나, 또 다른 작업 스레드에 명령 처리 요청을 한다. 예를 들면 GCC (Ground Command Control) 작업 스레드는 발사통제 장

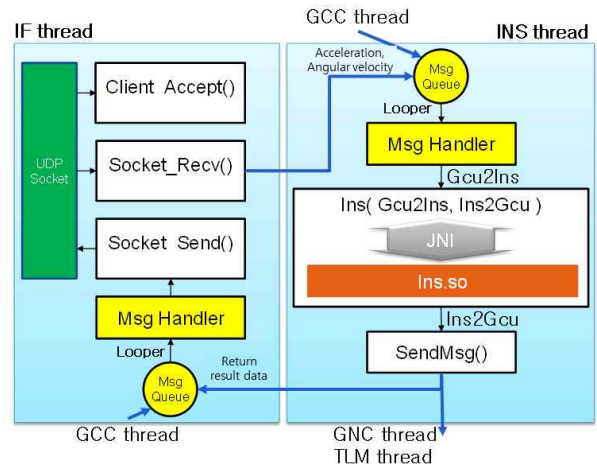


Fig. 7. Processing flow diagram of thread

비로부터 받은 명령의 처리에 대한 수행을 맡고 있다. 발사통제로부터 받은 명령을 해석하여 이를 직접 수행하거나, 이를 INS(Inertial Navigation System), GNC(Guidance and Control) 스레드로 전달하여 명령에 대한 처리를 수행토록 한다.

각각의 스레드는 서로 다른 스레드들과 Looper와 handler를 통해 Message 전달하고, 이 Message를 통해 서로 동기화하고 필요한 자원을 주고받는다. 스레드는 내부적으로 Looper를 가지며 그 안에는 Message Queue가 포함된다. Message Queue는 스레드가 다른 스레드나 혹은 자기 자신으로부터 전달받은 Message를 기본적으로 선입선출 형식으로 보관하는 Queue이다. Looper는 Message Queue에서 Message나 Runnable 객체를 차례로 꺼내 Handler가 처리하도록 전달한다.

작업 스레드와 인터페이스 스레드간의 관계도를 Fig. 7 에서 나타내었다. 인터페이스 스레드는 외부 장치로부터 받은 정보를 작업 스레드에 Message를 통해 전달하고, 작업 스레드는 알고리즘 수행결과를 인터페이스 스레드에 전달하여 외부장치로 결과 값을 리턴한다.

3.3 실시간 안드로이드 API

어플리케이션에 실시간성을 제공하기 위해서 RT-Android에서 제공하는 API를 살펴본다. 어플리케이션의 실시간성에 주된 영향을 미치는 파라미터들을 알아본다. RT-Android 연구팀은 소스 코드를 제공하지 않고, Raspberry-Pi를 포함한 몇몇 디바이스에서 이용 가능한 RT-Android의 롬 이미지와 실시간 기능 제어를 위한 API를 제공하고 있다.

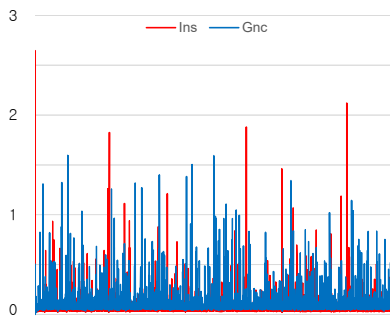


Fig. 8. The execution time of algorithm on Nexus 5x without real-time functionality (normalized w/r/t T)

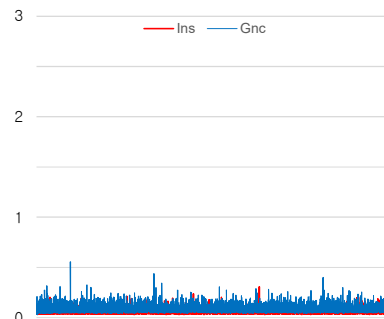


Fig. 9. The execution time of algorithm on Raspberry-pi with real-time functionality (normalized w/r/t T)

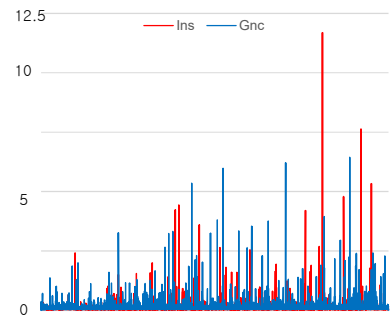


Fig. 10. The execution time of algorithm on Raspberry-pi without real-time functionality (normalized w/r/t T)

RTAndroid의 모든 속성은 RealTimeProxy 클래스를 통해 접근할 수 있다. 다음과 같이 RealTimeProxy 객체를 생성한 후 이 객체의 내부 함수를 통해 각각의 설정값들을 변경할 수 있다.

```
RealTimeProxy proxy = newRealTimeProxy();
```

첫 번째 속성값은 SchedulingPolicy 이다. 리눅스의 기본 scheduling policy인 Completely Fair Scheduler(SCHED_POLICY_NORMAL)를 포함하여 First-in - first-out (SCHED_POLICY_FIFO)과 SCHED_POLICY_RR(Round-robin)이 설정 가능하다.

```
proxy.setSchedulingPolicy(RealTimeConstants.SCHED_POLICY_FIFO);
```

Priority는 리눅스 커널에 의해 사용되는 프로세스의 우선순위로, 0부터 99까지 값이 사용되어진다. 높은 값일수록 우선순위가 높다.

```
int prio = 99;
proxy.setPriority(prio);
```

nice값은 한 프로세스가 다른 프로세스에 CPU time을 양보하도록 설정하는 값이다. -20부터 19까지의 값이 사용되며, 낮을수록 높은 우선순위를 가지게 된다.

```
proxy.setNiceValue(nice);
```

다음은 CPU의 Affinity로서, CPU가 다중 코어를 지닐 때, 실시간성을 필요로 하는 프로세스들을 특정 CPU 코어에서만 실행되도록 하여 실시간성을 높이는 방법이다.

```
int mask = (1<< cpuCore);
proxy.setAffinity(mask);
```

시뮬레이션에서, 위의 설정 값들에 의한 실시간 성능의 변화를 살펴본다.

IV. 시뮬레이션 및 성능 결과

Figure 8에서는 구현된 어플리케이션을 Nexus 5x 스마트폰위에서 실행했을 때 시간에 따른 항법 알고리즘과 유도조종 알고리즘의 수행시간을 보여준다. 그림에서 보이듯이 수행시간의 분포가 일정치 않고 매우 들쭉날쭉 하게 변화함을 볼 수 있다. 실험에서 센서 정보의 입력 주기에 따라 매번 알고리즘의 수행시간은 T 이내로 보장되어야 한다. 하지만 측정 결과로는 최대 5T 의 수행 시간 까지 보임을 볼 수 있다. 이는 안드로이드의 메모리 관리 프로세스 Garbage collection 프로세스가 알고리즘 수행 도중 작동하는 등, 커널과 자바 가상머신의 실시간성 보장 기능 부재가 원인으로 유추된다.

실시간성 지원에 따른 성능을 살펴보기 위해 같은 어플리케이션을 RT-Android가 올려진 Raspberry-Pi에서 실행하여 보았다. Fig. 9와 Fig. 10에서 실시간성 여부에 따른 유도조종 알고리즘의 수행시간을 나타내었다. 유도조종 알고리즘을 최우선적으로 실행하도록 실시간성을 조정하였을 때(Priority : 99, nice : -20, cpu affinity : 실시간 전용 CPU 코어 할당) 그렇지 않은 경우보다 평균 수행시간이 적을 뿐만 아니라 편차도 매우 적음을 알 수 있다.

Table 3을 보면 Raspberry-Pi에서 실시간성 지

Table 3. The execution time comparison of algorithm (normalized w/r/t T)

		Raspberry pi (RT on)	Raspberry pi (RT off)	Nexus 5x
Ins	Avr.	0.058	0.089	0.059
	Max	0.307	11.69	2.65
	Min	0.033	0.029	0.023
Gnc	Avr.	0.088	0.132	0.1
	Max	0.055	6.451	1.597
	Min	0.028	0.035	0.018

원이 적용 되었을 경우엔, Nexus 5x에서의 결과와 비교해서도 편차와 최대수행시간이 작음을 알 수 있다. 이는 Nexus 5x CPU (Hexa-core : 4x1.4GHz Cortex-A53, 2x1.8GHz Cortex-A57)의 성능이 Raspberry-Pi CPU(Quad-Core : 4x1.2GHz Cortex-A53)보다 뛰어난을 고려하면 놀라운 결과가 아닐 수 없다. 하지만 실험도중 종종 어플리케이션이 멈춰버리고, 디바이스가 아무런 반응을 하지 않는 프리징 현상이 발생해 실시간 안드로이드는 안정성을 확보하기까지는 좀 더 연구가 필요해 보인다.

V. 결 론

본 논문에서는 저가의 소형 유도탄 제어를 위한 스마트폰 활용을 제안하고, 이에 대한 시스템 구성과 유도조종 알고리즘을 수행할 유도조종방법 어플리케이션을 제안했다. 제안된 어플리케이션을 실제 Nexus-5x 스마트폰과 Raspberry-Pi에 올려 비행모의 시뮬레이션을 통해 성능을 분석했다.

무선 인터페이스, 자세제어 위치탐색 타깃 탐색들을 위한 다양한 센서들과 고성능 카메라, 그리고 이를 통합하여 처리할 수 있는 고성능의 프로세싱 능력들을 갖춘 스마트폰은 충분히 매력적인 기기이다. 하지만 한계점도 분명 존재한다. 개발된 어플리케이션의 스마트폰의 고성능 CPU에도 불구하고 실시간성 제한으로 일반적인 안드로이드로는 유도탄의 고속제어에서는 한계를 보이는 것으로 드러났다. 이에 따라 실시간성이 추가된 안드로이드나 스마트폰에서 이용 가능한 실시간 운영체제의 개발이 필수적이다. 본 논문에서는 다루지 않았지만, 추후 기기 내 센서 또는 외부장치와의 입출력 딜레이도 분석이 필요할 것으

로 보여진다. 스마트폰이 지닌 장점과 한계가 충분히 파악되고 분석될 때, 유도무기 또는 무기체계의 통합전자장치로의 활용도 충분히 실현 가능할 것으로 보여진다.

References

- 1) US Navy Locust demo vedio <https://youtu.be/AyguXoum3rk>
- 2) MBDA concepte visions - Vigilus 2012 <http://www.mbda-systems.com/innovation/concept-visions/vigilus-2012/>
- 3) PERDIX Fact sheet <https://www.defense.gov/Portals/1/Documents/pubs/Perdix%20Fact%20sheet.pdf>
- 4) Min J. Kim, Seol B. Bae, Moon G. Joo and Won Chang Lee, "Development of Android Smartphone Application for Mobile Robot Control," *Journal of Korean Institute of Information Technology*, Vol. 12, No. 5, 2014, pp. 7~13.
- 5) Jong-Min Kim, Young-Sub Kim, Eun Kim and Yun-Seok Lee, "A Study on the Drone Control System Using Android SmartPhone," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, Vol. 2015, No. 06, 2015, pp. 615~617.
- 6) Noh, J., Jung, H., Cho, W., Song, Y., "Development of Remote Controller for Multi-copter," *Proceedings of KSASS 2014*, 2014, pp. 837-840.
- 7) Seung-Hyun Yoon, Soon Mook Jung, Dong Won Choi, Kyung Min Lee, Key Ho Kwon, Jae Wook Jeon. "Remote Control of the Unmanned surveillance robot using Smartphone". *HCI 2012*, 2012, pp. 50-52.
- 8) Yongxun Quan, Hyunsik Ahn. "Android Based Ubiquitous Interface for Controlling Service Robots," *The Journal of the Institute of Internet, Broadcasting and Communication*, Vol. 10, No. 3, 2010, pp. 35-41.
- 9) Loianno, G., Cross, G., Qu, C., Mulgaonkar, Y., Hesch, J. A., and Kumar, V., "Flying smartphones: Automated flight enabled by consumer electronics," *IEEE Robotics & Automation Magazine*, Vol. 22, No. 2, 2015, pp. 24-32.
- 10) Loianno, G., Mulgaonkar, Y., Brunner,

C., Ahuja, D., Ramanandan, A., Chari, M., Diaz, S., and Kumar, V., "A swarm of flying smartphones," *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, IEEE, 2016, pp. 1681-1688.

11) Bryant, P., Gradwell, G., and Claveau, D., "Autonomous UAS controlled by onboard smartphone," *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, IEEE, 2015, pp. 451-454.

12) Faragher, R. M., Chick, O. R., Wagner, D. T., Goh, T., Snee, J., and Jones, B., "Captain Buzz: An All-Smartphone Autonomous Delta-Wing Drone," *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ACM, 2015, pp. 27-32.

13) Kim, J., Lee, D., Shim, H., "Vision-based Target Detection and Geo-location System for UAVs using a Smartphone," *Proceeding of KSASS 2013*, 2013, pp. 457-460.

14) Salas, A. G., Attai, W., Oyadomari, K. Y., Priscal, C., Schimmin, R. S., Gazulla, O. T., and Wolfe, J. L., "Phonesat in-flight experience results," *Proceedings of the Samll Satelites and Services Symposium 2014*, 2014.

15) Guo, J., Chu, J., and Gill, E., "Onboard autonomy of CubeSat clusters based on smartphone technology," *5th International Conference on Spacecraft Formation Flying Mission and Technologies*, Munich, 2013.

16) Kenyon, S., Bridges, C., Liddle, D., Dyer, R., Parsons, J., Feltham, D., Taylor, R., Mellor, D., Schofield, A., and Linehan, R., "STRaND-1: Use of a \$500 Smartphone as the Central Avionics of a Nanosatellite," *Proceedings of the 2nd International Astronautical Congress 2011*, 2011.

17) Kalkov, I., Franke, D., Schommer, J. F., and Kowalewski, S., "A real-time extension to the Android platform," *Proceedings of the 10th international workshop on Java technologies for real-time and embedded systems*, ACM, 2012, pp. 105-114.

18) Gerlitz, T., Kalkov, I., Schommer, J.,

Franke, D., and Kowalewski, S., "Non-Blocking Garbage Collection for Real-Time Android," *Proceeding of 11th International Workshop on Java Technologies for Real-time and Embedded Systems*, 2013, ACM, pp. 108-117.

19) Real-Time Linux Wiki.

https://rt.wiki.kernel.org/index.php/Main_Page.

20) Kang, H., Kim, D., Kang, J., and Kim, K., "Real-time motion control on Android platform," *The Journal of Supercomputing*, Vol. 72, No. 1, 2016, pp. 196-213.

21) Kolek, K., "Application of Android OS as real-time control platform," *In Automatyka / Automatics*, Vol. 17, No. 2, 2013.

22) Maia, C., Nogueira, L. M., and Pinho, L. M., "Evaluating android os for embedded real-time systems," *6th International Workshop on Operating Systems Platforms for Embedded Real-Time Applications*, 2010, pp. 63-70.

23) Lee, S. and Jeon, J. W., "Evaluating performance of Android platform using native C for embedded systems," *Control Automation and Systems (ICCAS), 2010 International Conference on*, IEEE, 2010, pp. 1160-1163.

24) Nong, L., Wang, L., and Huang, Y., "Application research of android in embedded vehicle navigation system," *Computer Engineering and Design*, Vol. 31, No. 11, 2010, pp. 2473-2476.

25) Yan, Y., Konduri, S. H., Kulkarni, A., Anand, V., Ko, S. Y., and Ziarek, L., "RTDroid: a design for real-time Android," *Proceedings of the 11th International Workshop on Java Technologies for Real-time and Embedded Systems*, ACM, 2013, pp. 98-107.

26) Dohyeon Kim, Hyeongseok Kang, Jeongnam Kang, Eungyu Lee and Kanghee Kim, "Design and Implementation of an Android Application for Real-time Motion Control," *KIISE Transactions on Computing Practices*, Vol. 21, No. 4, 2015, pp. 315~319.

27) Syntronics DSALS Presentation

<http://www.syntronics.net/assets/files/cutsheets/DSALS%20Presentation%20Short%20Public.pdf>