

항공기 3차원 충돌회피 알고리즘 구현과 실시간 운영체계를 이용한 Micro Controller Unit의 성능 비교

임지성¹ · 김동신² · 박인혁² · 이상철^{2,†}

¹한국항공대학교 대학원 항공우주 및 기계공학과

²한국항공대학교 항공우주 및 기계공학부

Implementation of 3-D Collision Avoidance Algorithm and Comparison of Micro Controller Unit's Performance using Real-Time Operating System

Ji-Sung Lim¹, Dong-Sin Kim², In-Hyeok Park² and Sangchul Lee^{2,†}

¹Graduate School of Aerospace and Mechanical Engineering, Korea Aerospace University

²School of Aerospace and Mechanical Engineering, Korea Aerospace University

Abstract

In this study, Real-Time Operating System(RTOS) and 3-D collision avoidance algorithm are implemented to three different Micro Controller Unit(MCU)s and their performances compared. We selected Microchip Technology's ATmega2560, STM's ARM Cortex-M3 and ARM Cortex-M4, because they are widely used. FreeRTOS, an open-source operating system, was also used. The 3D collision avoidance algorithm consists of the vertical and the horizontal avoidance algorithm, which is implemented using C++. The performances of the MCUs were compared with respect to used memory and calculation time. As a result, Cortex-M4's calculation time was the fastest and ATmega2560 used least memory.

초 록

본 논문에서는 RTOS과 항공기의 3차원 충돌회피 알고리즘을 세 개의 MCU에 적용하여 각 MCU의 성능을 비교하였다. MCU는 많이 사용되는 Microchip Technology사의 ATmega2560과 STM사의 ARM Cortex-M3, ARM Cortex-M4를 선정하였으며, RTOS는 공개되어 있는 FreeRTOS를 사용하였다. 성능을 확인하기 위해 적용된 3차원 충돌회피 알고리즘은 수직회피와 수평회피를 통합한 알고리즘이며 C++로 구현하였다. MCU의 성능은 각 MCU의 사용 메모리와 계산 시간을 측정하여 비교하였다. 비교 결과 세 MCU 중, 계산 시간은 ARM Cortex-M4가 빨랐으며, ATmega2560이 적은 메모리를 사용하였다.

Key Words : 3-D Collision Avoidance Algorithm(3차원 충돌회피), Real Time Operating System(실시간운영체제), Micro Controller Unit(마이크로 컨트롤러 유닛)

1. 서 론

현대 항공기의 복잡하고 나날이 증가되는 임무 요구사항을 수행하기 위해, 항공전자시스템 기술발전의 비중이 증가되고 있다[1-3]. 이에 개발기간 단축과 개발비용 절감을 위해, 항공전자시스템의 개발에 상용기성품 적용가능성에 대한 연구가 진행되었다[2,3]. 또한,

복잡한 임무 요구사항을 실시간으로 정확하고 효율적으로 관리하기 위한 실시간 운영체제(RTOS; real-time operating system)의 개발도 활발히 이루어지고 있다. RTOS는 시간에 따른 정확한 태스크 관리, 시스템 자원 관리가 필요한 실시간 응용프로그램을 위해 개발된 운영체제이다.

본 논문에서는 실시간 운영체계를 이용한 상용기성품 마이크로 컨트롤러 유닛 (MCU; micro-controller unit)의 성능을 비교하였다. MCU 성능 비교를 위해 수직 충돌회피 알고리즘과 수평 충돌회피 알고리즘을 통합하여 3차원 충돌회피 알고리즘을 C++언어로 구

현하였다[4]. 실시간 운영체제는 오픈소스인 FreeRTOS를 사용하였다. MCU는 8Bit 프로세서인 ATmega2560, 32Bit 프로세서인 ARM Cortex M3, ARM Cortex M4를 사용하였다. 구현한 알고리즘 실행과일을 이용하여 FreeRTOS가 탑재된 각 MCU의 계산시간과 메모리 사용량을 측정하여 MCU의 성능을 비교하였다.

2. 본 론

2.1 3차원 충돌회피 알고리즘

Figure 1은 3차원 충돌회피 알고리즘을 구현하는데 사용된 NED(North East Down) 좌표계이다[4,5]. 항공기의 선속도 성분과 항공기 선회반경은 Eq. (1)에서 Eq. (4)와 같이 표현할 수 있다[6].

$$\dot{x} = V \cos \gamma \cos \chi \quad (1)$$

$$\dot{y} = V \cos \gamma \sin \chi \quad (2)$$

$$\dot{z} = V \sin \gamma \quad (3)$$

$$R = \frac{V^2}{g \tan \phi} \quad (4)$$

여기서, V 는 항공기의 속도, γ 는 비행경로 각, χ 는 방위각, R 은 항공기의 선회반경, ϕ 는 항공기 뱅크 각, g 는 중력 가속도를 각각 의미한다.

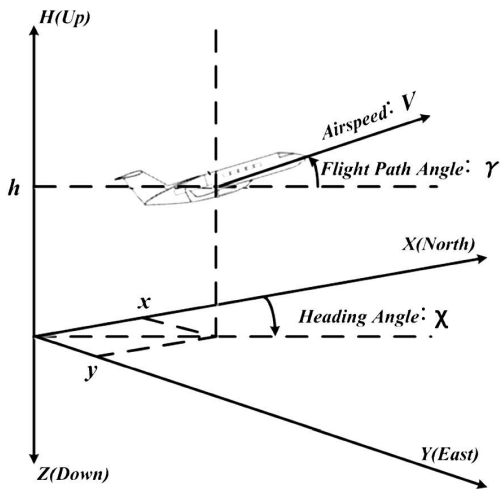


Fig. 1 Coordinate System

Equation (1)부터 Equation (4)에서 항공기의 비행경로 각과 비행속도가 일정하다는 가정 하에, 항공기가 선회를 t 시간 동안 수행했을 때의 항공기의 위치는 Eq. (5)와 같이 표현할 수 있다[6].

$$P = [x, y] = [x_0 + R \operatorname{sgn}(\Delta \chi) \cdot (\cos(\chi) - \cos(\chi + \Delta \chi)), y_0 + R \operatorname{sgn}(\Delta \chi) \cdot (\sin(\chi) + \sin(\chi + \Delta \chi))] \quad (5)$$

여기서, x_0, y_0 는 선회를 시작할 때의 항공기의 위치, sgn 은 부호함수, $\Delta \chi = tg \tan \phi / V$ 를 각각 의미한다.

3차원 충돌회피 알고리즘은 수직회피 알고리즘과 수평회피 알고리즘을 통합하는 방법을 사용하였다[6]. 수직 충돌회피 알고리즘은 항공기에 적용되고 있는 공중충돌방지장치 (TCAS; traffic alert and collision avoidance system)를 기준으로[7,8], 수평 충돌회피 알고리즘은 Andrew Trapani, Heinz Erzberger가 발표한 연구를 기반으로 구현하였다[9,10]. Figure 2는 구현된 3차원 충돌회피 알고리즘 순서도를 보여준다. Figure. 2에서 볼 수 있듯이 항공기의 충돌확률 P_C 가 기준 값 P_C^* 보다 크고, 최근접 지점까지의 남은시간 τ 가 기준 값 τ^* 보다 작으면 3차원 충돌회피를 지시하게 된다. 이와 반대되는 경우, 항공기가 현재 비행경로를 유지하도록 지시하게 된다. Table 1은 구현된 3차원 충돌회피 알고리즘의 특성을 보여준다.

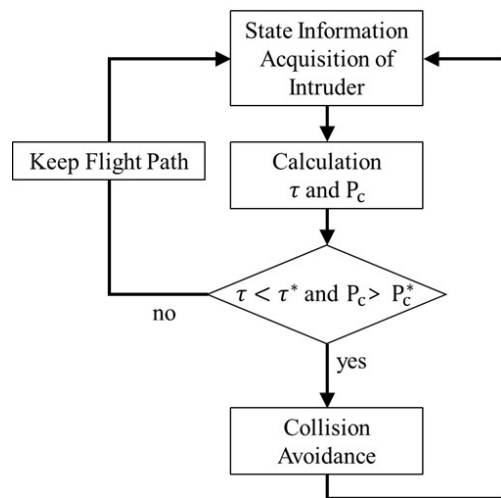


Fig. 2 3-D Collision Avoidance Algorithm Flow Chart

Figure 2에서의 항공기의 충돌확률 P_C 는 최근접 지점에서의 각 축에 대한 두 항공기의 상대거리(r_f)를 이용해 구할 수 있다. 두 항공기의 위치와 속도는 불확실성을 갖는 랜덤변수이므로, r_f 는 랜덤확률분포를 가지는 랜덤변수이고, σ_{r_f} 는 r_f 의 분산으로 정의할 수 있다. Equation (6)은 σ_{r_f} 와 r_f 를 이용해 충돌확률 P_C 를 구하는 것을 보여준다[6].

$$P_C = \frac{1}{\sqrt{2\pi} \sigma_{r_f}} \int_{-A}^A \exp\left(-\left(\frac{r_f - \bar{r}_f}{\sqrt{2} \sigma_{r_f}}\right)^2\right) d_{r_f} \quad (6)$$

Table 1 3-D collision avoidance algorithm characteristics

상태변수 추정 방법	Probabilistic
상태변수 차원	Horizontal and Vertical
충돌회피 방법	Prescribed
회피 기동	Horizontal and Vertical
다중충돌 회피 방법	Pairwise
충돌 회피의 책임	Cooperative

2.2 RTOS

RTOS는 태스크 단위로 기능을 수행하며, 각 태스크는 우선순위를 가지고 있어 높은 우선순위의 태스크가 먼저 실행되며 시작 및 종료시점의 변경이 가능하다. 또한, IPC(interprocess communication)메커니즘을 통하여 태스크들 간의 통신 및 원활한 동기화를 제공한다[11]. 이러한 특징으로, RTOS는 항공기, 자동차, 철도, 로봇, 우주선 제어 및 무기 통제, 기상 관측, 가상 현실 등의 분야에 적용될 수 있다[12]. 특히, 항공우주 분야로는 무인기 자동항법장치, 전투기 및 훈련기 장비, 각종 유도탄 유도장치, 레이더 장치, 센서 네트워크 장치, 정찰 로봇, 어뢰기 등에 적용되고 있다.

상용 RTOS로는 VxWorks, INTEGRITY, OSE RTOS, QNX RTOS, LynxOS 등이 있으며, VxWorks의 경우

국방 및 항공우주분야에 특화된 상용 RTOS이자 최고 점유율을 차지하고 있다[13]. 오픈소스 RTOS로는 uC/OS II, FreeRTOS, RTEMS 등이 있으며, 본 논문에서는 FreeRTOS를 사용하였다. FreeRTOS는 4000 라인 내에서 8 bit, 16 bit, 32 bit 의 다양한 MCU에 포팅할 수 있으며, Multitasking, Co-routine, Queue, Semaphore 등의 기능을 지원한다[14].

2.3 3차원 충돌회피 시뮬레이션 결과

3차원 충돌회피 알고리즘은 C 언어를 기반으로 구현하였다. 3차원 충돌회피 알고리즘의 시뮬레이션 결과를 확인하기 위해 Fig. 3 및 Fig. 4와 같이 항공기가 수직으로 횡단하는 상황과 마주보며 오는 상황을 가정하였다. 수직으로 횡단하는 상황을 scenario 1, 마주보며 오는 상황을 scenario 2로 정의하였다. 각 상황의 초기 조건은 Table 2 및 Table 3과 같다. 시뮬레이션 결과는 두 항공기 A, B간의 거리, 이동경로, 고도로 확인하였다. 3차원 충돌회피 알고리즘의 시뮬레이션 결과는 Fig. 5에서 Fig. 12와 같다.

Table 2 Initial condition of scenario 1

	항공기 A	항공기 B
항공기 초기위치	(0, 0, 1) nm	(12, 12, 1) nm
속도	300 knot	380 knot
상승/하강 속도 (수직회피 시)	1500 ft/min	
회피 뱅크 각 (수평회피 시)	20°	
헤딩	0°	270°

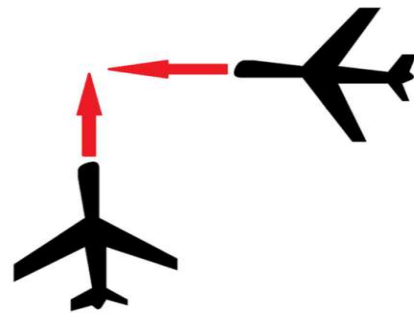


Fig. 3 Scenario 1 of pair of Aircraft

Table 3 Initial condition of scenario 2

	항공기 A	항공기 B
항공기 초기위치	(-12, 12, 1) nm	(12, 12, 1) nm
속도	380 knot	380 knot
상승/하강 속도 (수직회피 시)	1500 ft/min	
회피 뱅크 각 (수평회피 시)	20°	
헤딩	90°	270°

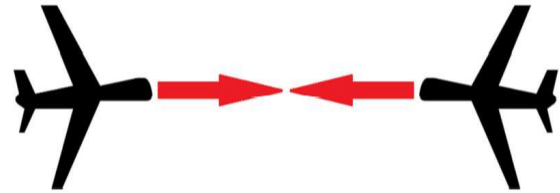


Fig. 4 Scenario 2 of pair of Aircraft

Figure 5는 횡단하는 항공기(Scenario 1)의 3차원 시물레이션 결과이다. Figure. 6 및 Figure. 7을 통해 3차원 충돌회피 알고리즘에서 설계된 것과 같이 고도회피와 수평회피를 동시에 수행하는 것을 볼 수 있다.

Figure 8는 마주보는 항공기(Scenario 2)의 3차원 시물레이션 결과이다. Figure. 9 및 Figure. 10을 통해 3차원 충돌회피 알고리즘에서 설계된 것과 같이 고도회피와 수평회피를 동시에 수행하는 것을 볼 수 있다.

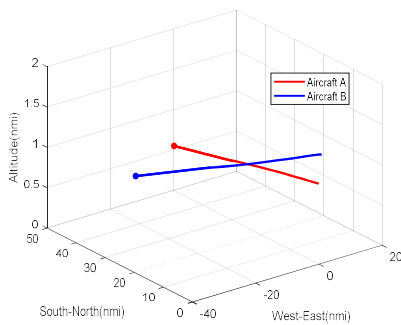


Fig. 5 3-D view (Scenario 1)

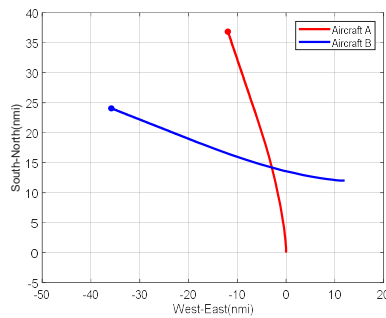


Fig. 6 Top view (Scenario 1)

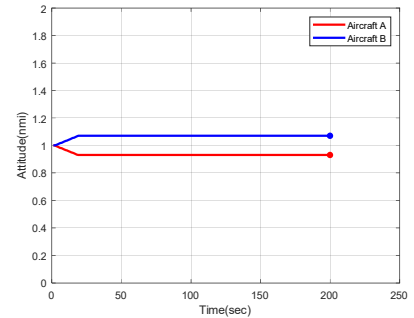


Fig. 7 Altitude (Scenario 1)

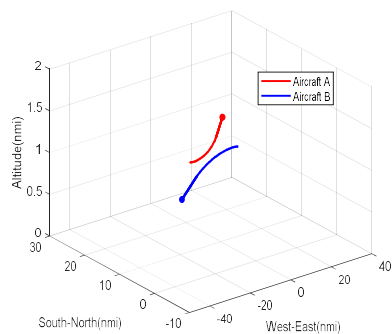


Fig. 8 3-D view (Scenario 2)

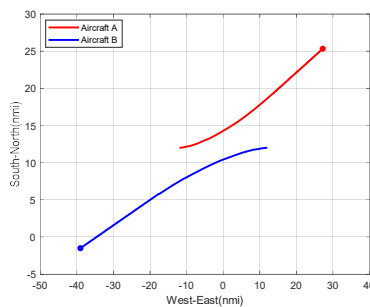


Fig. 9 Top view (Scenario 2)

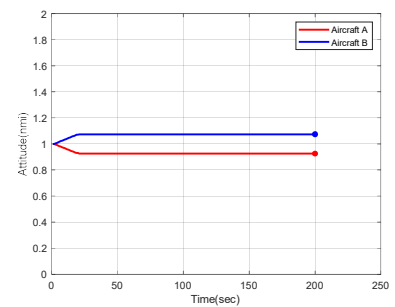


Fig. 10 Altitude (Scenario 2)

2.4 Micro Controller Unit 성능 결과

본 논문에서 사용된 MCU는 Microchip Technology사의 ATmega 2560과 STM사의 ARM Cortex-M3(STM32F103RBT6), ARM Cortex-M4(STM32F407VG T6)이며, 세 MCU의 스펙은 Table 4와 같다[15-16]. 세 MCU의 성능 확인을 위해 적용한 3차원 충돌회피 알고리즘은 서로 같으며, 각 MCU에서 실행과일이 구동되기 위한 명령어만 조금 상이하다. 성능확인을 위해 수집한 파라미터는 계산시간과 사용된 메모리이며, 각각의 MCU에서 구동하였을 때 결과는 Table 5에서 Table 7과 같다.

Table 5는 각 MCU에서 사용된 메모리를 보여준다. 각 MCU내에서 Scenario 1과 Scenario 2를 적용했을 때, 같은 메모리를 사용하는 것을 확인할 수 있었다. Table 5를 보면 ARM Cortex-M3와 ARM Cortex-M4의 메모리사용량은 차이가 없으며, ARM Cortex-M3와 ARM Cortex-M4이 ATmega2560보다 약 1.5배 메모리를 많이 사용한다는 것을 알 수 있다. 이는 32 bit 프로세서가 8bit 프로세서에 비해 명령어 집합 (Instruction Set)이 더 많고, 동일한 변수더라도 자료형(Data Type)이 더 크며, 메모리 주소(Memory Address) 구성이 더 복잡하기 때문이라고 판단된다.

Table 4 Specification of MCU

	ATmega 2560	Cortex-M3	Cortex-M4
Data bus	8 bit	32 bit	32 bit
Flash Memory	256 KB	128 KB	1024 KB
RAM	8 KB	20 KB	192 KB
Clock speed	16 MHz	64 MHz	168 MHz

Table 5 Used memory of each MCU

	ATmega2560	Cortex-M3	Cortex-M4
3차원 회피 알고리즘 (scenario 1, scenario 2)	3885 byte	5943 byte	5937 byte

Table 6 및 Table 7 은 각각 항공기가 회피를 수행하는 동안 실시간으로 위치데이터를 출력하는 경우와,

출력하지 않는 경우에 따른 각 MCU의 계산시간을 보여준다. Table 6과 Table 7을 비교해보면, 위치데이터를 출력하는데 걸리는 시간은 각각 ATmega2560 약 4.8초, ARM Cortex-M3 약 3.6초, ARM Cortex-M4 약 3.65초 이다.

Scenario 1과 Scenario 2 모두 ARM Cortex-M4가 계산시간이 빠르다는 것을 확인할 수 있다. 위치출력을 고려하지 않고, 단순 계산시간만을 비교해봤을 때 ARM Cortex-M3를 기준으로 ARM Cortex-M4는 약 3.3배 빠르고, ATmega2560은 약 3.5배 느리다는 것을 확인할 수 있다. 이는 Clock speed가 제일 높은 ARM Cortex-M4가 계산시간이 적게 걸렸으며, Clock speed가 가장 낮은 Atmega 2560이 계산 시간이 오래 걸린 것으로 판단된다.

Table 6 Process time of each Micro Controller Unit(위치데이터 출력하는 경우)

	ATmega 2560	Cortex-M3	Cortex-M4
3차원 회피 알고리즘 (scenario 1)	76.4 sec	23.9 sec	9.7 sec
3차원 회피 알고리즘 (scenario 2)	35.3 sec	12.3 sec	6.2 sec

Table 7 Process time of each Micro Controller Unit(위치데이터 출력하지 않는 경우)

	ATmega 2560	Cortex-M3	Cortex-M4
3차원 회피 알고리즘 (scenario 1)	71.6 sec	20.3 sec	6.0 sec
3차원 회피 알고리즘 (scenario 2)	30.5 sec	8.7 sec	2.6 sec

3. 결 론

본 논문에서는 실시간 운영체제와 MCU를 이용하여 항공기의 3차원 충돌회피 알고리즘을 구현하였고 MCU 성능을 확인하였다. 사용된 MCU는 흔히 사용되는

Microchip Technology사의 ATmega2560, STM사의 ARM Cortex-M3, ARM Coretex-M4이고, 실시간 운영체계는 무료로 제공되고 있는 FreeRTOS를 사용하였다. 항공기 3차원 충돌회피 알고리즘은 C++ 를 기반으로 구현하였고, 동일한 실행 파일과 RTOS를 이용하여 각각의 MCU에서의 계산시간과 사용된 메모리 값을 측정하였다. 3차원 충돌회피 알고리즘을 각각의 MCU에서 실행한 결과 거의 동일한 시뮬레이션 결과가 출력되었다. MCU의 성능비교 결과로는 ARM Cortex-M3, ARM Cortex-M4 성능은 ATmega2560 보다 계산 시간은 약 3.5배, 약 12배 빠르며, 약 1.5배 많은 메모리를 사용하는 것을 확인할 수 있었다.

References

- [1] C. G. Park, "Design of Lightweight RTOS for MCU," *The journal of the Korea Institute of Maritime Information & Communication Sciences*, Vol.15, No.6, pp.1301-1306, 2011.
- [2] I. Moir, A. Seabrides, AIAA Eudation Series Aircraft Systems, *AIAA*, pp. 315-337, 2001.
- [3] S. C. Lee, I. K. Kim and Y. I. Kim, "Development of Operational Flight Program for Stores Management Computer," *Journal of the Korean society for aeronautical & space sciences Journal*, Vol.31, No. 5, pp.124-133, 2003.
- [4] S. W. Yang and S. C. Lee, "A Study on the Development of an Avionics System," *Journal of the Korean Society for Aviation and Aeronautics Journal*, Vol.15, No.1, pp.61-67, 2007.
- [5] M. K. Park, K. B. Kim, and K. J. Yoon, "Design and Fabrication of coaxial quadrotor for indoor flight," *KSAS 2012 fall conference*, pp. 1837-1840, November, 2012.
- [6] Y. R. Kim, "A Development and Verification of 3-D Resolution Algorithm for Aircraft Collision Avoidance," Master Degree Thesis, Korea Aerospace University.2012
- [7] B. K. Jun, "Performance Improvement of the Collision Avoidance Algorithm for the TCAS-II System with Safety Critical Software," Master Degree Thesis, Korea Aerospace University. 2011.
- [8] J. K. Kuchar and A. C. Drumm, "The Traffic Alert and Collision Avoidance System," *Lincoln Laboratory Journal*, Vol.16, No2, 2007.
- [9] A. Trapani, H. Erzberger and W. Dunbar, "Performance Analysis of a Horizontal Separation Assurance Algorithm for Short-Range Conflicts," *AIAA Guidance, Navigation, and Control Conference*, Aug, 2009.
- [10] A. Trapani, "Performance Analysis of a Horizontal Separation Assurance Algorithm for Short-Range Conflicts Detection and Resolution," University of California Santa Cruz, Dec, 2008.
- [11] S. H. Kim, S. O. Cho, S. S. Kim, C. K. Ryoo and K. Y. Choi, "Development of Embedded Program for UAV Flight Control System using RTOS and Model-Based Auto Code Generation," *Journal of Korean Society for Aeronautical and Space Science*, Vol.39, No.10, pp.979-986, 2011.
- [12] C. H. Koo, "Real-time software Development Technology Trends," *Current Industrial and Technological Trends in Aerospace*, Vol.2, No.1, pp.86-93, 2004.
- [13] S. J. Lee and D. H. Kim, "Technology Trends of Real-Time OS for Defense Application," *Communications of the Korean Institute of Information Scientists and Engineers*, Vol.29, No.10, pp.41-51, 2011.
- [14] B. J. Hong, "A Design on CAN Communication Security on FreeRTOS," Master Degree Thsis, Soongsil University, pp.4-7, 2017.
- [15] Atmega 2560 Device Overview, <https://www.microchip.com/wwwproducts/en/ATmega2560>
- [16] Cortex M3 and M4 Product Specification, https://www.st.com/content/st_com/en/search.html?q=CO RTEX%20M3-t=products-page=1