Journal of the KIMST, Vol. 21, No. 3, pp. 370-378, 2018
ISSN 1598-9127
DOI http://dx.doi.org/10.9766/KIMST.2018.21.3.370

Research Paper 정보 · 통신 부문

MIL-STD-1553B 통신에서 샘플링 기반 최적화 기법을 이용한 효율적 임무 자료 전송

이헌철*,1) · 김기표1) · 권용성1)

1) 국방과학연구소 제1기술연구본부

Efficient Mission Data Transmission with Sampling-Based Optimization in MIL-STD-1553B

Heoncheol Lee*,1) · Kipyo Kim1) · Yongsung Kwon1)

1) *The 1st Research and Development Institute, Agency for Defense Development, Korea*

(Received 13 October 2017 / Revised 10 January 2018 / Accepted 13 April 2018)

ABSTRACT

The mission data in missile systems should be quickly and reliably transmitted from a mission transmission device to a guidance control unit. The MIL-STD-1553B is one of the reliable communication standards, but its bit rate is generally limited to 1Mbps due to the intrinsic properties of its electrical design. Therefore, the bus controller needs to be optimized to efficiently transmit the mission data on the inevitably limited bit rate. This paper proposes an analytical approach based on sampling-based optimization methods to maximize the data throughput without data loss. The proposed approach was evaluated in the simulations with the data transmission model for the MIL-STD-1553B communication system. The results of the proposed methods were applied to a real-time system and showed that the proposed method was successfully performed.

Key Words : MIL-STD-1553B, Mission Data Transmission(임무 자료 전송), Sampling-based Optimization(샘플링 기반 최적화)

1. 서 론

항공우주 및 국방 분야, 특히 유도무기 분야와 같이 높은 신뢰도가 요구되는 시스템에서 MIL-STD-1553B 통신이 널리 사용되고 있다[1]. 유도무기 시스템의 경우, 임무 자료(mission data)는 Fig. 1과 같이 일반적으로 발사 전에 버스 제어기(Bus Controller, BC) 역할을 수행하는 임무 전송 장비(mission data transmission device)로부터 원격 단말기 역할(Remote Terminal, RT)을 수행하는 유도제어장치(Guidance Control Unit, GCU)로 전송된다.

* Corresponding author, E-mail: heonclee@add.re.kr
Copyright ⓒ The Korea Institute of Military Science and Technology

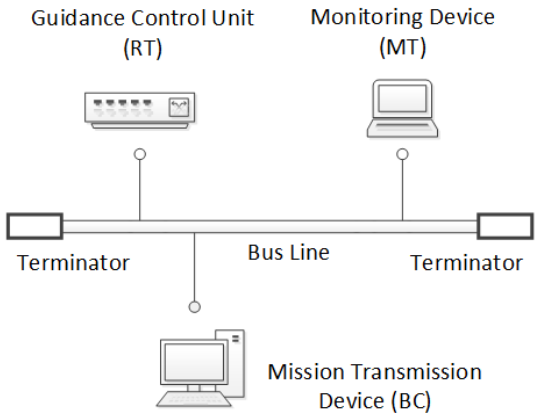


Fig. 1. The configuration of MIL-STD-1553B

최근 고성능 정밀타격을 위해 첨단 센서 장비를 사용함에 따라 임무 자료의 용량이 점점 증가되고 있다. 하지만, MIL-STD-1553B 통신은 설계상 전기적 특성으로 인해 비트 전송률(bit rate)이 1 Mbps로 제한되기 때문에, 대용량 임무 자료 전송을 위해서는 버스 제어기 운용의 효율성을 높여야 한다. 최근 이와 관련하여 다양한 연구들이 진행되어 왔다[3-5]. MIL-STD-1553B 통신에서 일반적으로 임무 자료는 버스 제어기에서 다중메시지체인을 사용하고, 원격 단말기에서 더블 버퍼를 사용함으로써 보다 효율적으로 전송될 수 있다[2].

본 연구에서는 다중메시지체인 및 더블버퍼링 기술이 적용된 MIL-STD-1553B 통신에서, 다중메시지체인을 구성하는 메시지의 개수와 전송 주기에 주목한다. 메시지의 개수와 전송 주기의 비율은 자료 처리율(data throughput)을 결정하게 된다. 자료 처리율이 높을수록 동일 시간 내에 전송할 수 있는 데이터의 양이 많아지지만, 메시지 개수가 과도하게 증가하게 되면 다음 전송 주기까지 유도제어장치에 전송된 데이터 처리의 미완료로 인한 임무 자료 손실이 발생할 수 있다. 따라서 본 논문에서는 다중메시지체인 및 더블버퍼링 기술이 적용된 MIL-STD-1553B 통신에서, 샘플링 기반 최적화(sampling-based optimization) 기법들을 이용하여 다중메시지체인을 구성하는 최적 메시지의 개수와 최적 전송 주기를 계산할 수 있는 방법을 제안한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장에서는 다중메시지체인 및 더블버퍼링 기술이 적용된 MIL-STD-1553B 기반 임무 자료 전송 모델을 설명한다. 3장에서는 샘플링 기반 최적화 기법들을 이용한 최적 메시지의 개수와 최적 전송 주기를 획득하는 방법을 제안한다. 4장에서는 제안된 방법의 성능을 평가하고, 실제 시스템에 적용된 결과를 나타낸다. 마지막으로, 5장에서는 본 논문의 결론을 맺는다.

2. MIL-STD-1553B 기반 임무 자료 전송 모델

본 장에서는 먼저 다중메시지체인 및 더블버퍼링 기술이 적용된 MIL-STD-1553B 기반 임무 자료 전송 모델을 설명한 후, 효율성을 높이기 위한 최적화 문제를 다룬다.

2.1 MIL-STD-1553B 기반 임무 자료 전송

MIL-STD-1553B 통신 시스템은 워드(word) 또는 메시지(message) 형식의 자료를 송수신하는 직렬(serial) 버스 시스템이다. 하나의 워드는 3개의 동기(sync) 비트(bit), 16개의 자료 또는 명령 및 상태 비트, 그리고 1개의 패리티(parity) 비트로 구성되어 있다. 하나의 메시지는 적어도 하나의 명령 워드를 포함한 워드들의 집합이다[2]. 모드 명령용 메시지를 제외하고는, 일반적으로 한 메시지는 적어도 1개에서 32개의 자료 워드들을 포함하게 된다. 메시지의 크기는 통신 프로토콜과 함께 시스템 요구조건에 의해 정의된다. 버스 제어기에서 송신되는 메시지들은 IMGT(Inter-Message Gap Time)에 의해 구분된다. 메시지체인(message chain)은 여러 개의 메시지들로 묶여진 그룹이다. 메시지체인을 구성하는 메시지들의 서브-주소(sub-address)가 동일하면 단일메시지체인(Single Message Chain, SMC), 다르면 다중메시지체인(Multiple Message Chain, MMC)이라 한다.

Kim[6] 등은 버스 제어기에서 SMC를 사용했을 때와 MMC를 사용했을 때, 그리고 원격 단말기에서 단일 버퍼를 사용했을 때와 더블 버퍼를 사용했을 때의 자료 처리율을 비교 및 분석하였다. 그들의 결과에서는 MMC와 더블 버퍼를 사용했을 때 가장 높은 자료 처리율을 보였다. 하지만, 그들의 연구에서는 버스 제어기에서 MMC를 구성할 때 사용되는 메시지의 개수와 MMC의 전송 주기를 경험적(heuristic) 방법을 이용하여 결정했다. 본 논문에서는 MIL-STD-1553B 통신 시스템에서 MMC와 더블 버퍼를 사용함에 있어서, MMC를 구성하는 최적 메시지 개수와 최적 전송 주기를 찾을 수 있는 방법을 제안하고자 한다.

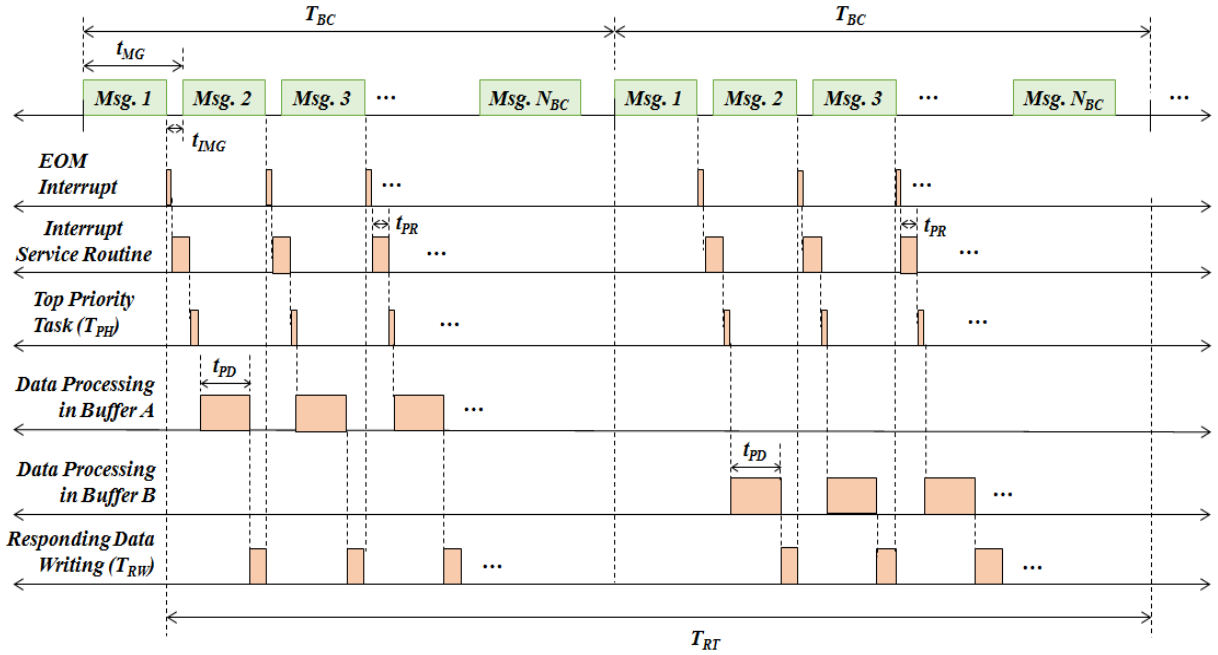


Fig. 2. The timing diagram for MIL-STD-1553B with MMC and double buffers

MMC를 사용하는 버스 제어기와 더블 버퍼를 사용하는 원격 단말기 간 통신 타이밍 다이어그램(timing diagram)은 Fig. 2와 같다. 각 전송주기, T_{BC}마다 버스 제어기는 N_{BC} 개의 메시지들로 구성된 MMC를 원격 단말기로 전송한다. 각 메시지들은 IMGT, 즉 t_{IMG}에 의해 구분되고, 하나의 메시지 전송에 걸리는 시간인 t_{MG}는 t_{IMG}를 포함한 시간이다. 원격 단말기에선 먼저 MIL-STD-1553B 칩에서 발행된 EOM(End-Of-Message) 인터럽트를 처리하기 위해, 인터럽트 서비스 루틴(Interrupt Service Routine, ISR)을 실행하기 위한 시간(t_{PR})이 필요하다. 또한 최상위 우선순위 태스크 처리를 위한 시간(T_{PH})이 필요하다, 그리고 수신한 MMC의 각 메시지를 처리하기 위한 시간(t_{PD})과 응답 버퍼에 응답 메시지를 쓰기 위한 시간(T_{RW})이 필요하다. 원격 단말기에서는 다음 MMC 수신 전에 현재 수신한 MMC에 대한 처리가 완전히 종료되어야만 자료 손실이 발생하지 않는다.

2.2 최적화 기법을 통한 문제 해결

만약 원격 단말기에서 요구되는 시간 T_{RT}가 버스 제어기에서의 전송 주기 T_{BC}의 두 배보다 크게 되면, 원격 단말기에서 현재 처리 중인 MMC가 담겨있는 버퍼의 전부 혹은 일부를 다음 MMC에 의해 덮어쓰게 되기 때문에 자료 손실이 발생하게 된다. 즉, 자료 전송의 신뢰성이 저하된다. 이 문제를 해결하기 위해 단순히 T_{BC}를 증가시키는 것은 자료 처리율의 저하를 발생시키기 때문에 효율성 면에서 좋은 방법이 아니다. 따라서 자료 전송의 신뢰성을 유지하면서 효율성을 높일 수 있는 적절한 T_{BC}와 N_{BC}를 찾는 것은 어려운 문제이다.

본 논문에서는 이 문제를 최적화 관점에서 접근함으로써 해결하고자 한다. 다양한 최적화 기법들이 있고 각자 장단점을 가지고 있다. 다만, 본 연구와 같이 최적화 대상의 모델이 비선형인 경우, 즉 제약조건(constraint) 수식들이 비선형인 경우에는 편미분 수식을 유도하기 힘들기 때문에 전통적인 최적화 기법들은 적용하기 힘들다. 따라서 본 논문에서는 샘플링 기반 최적화 기법을 이용하여 임무 자료 전송의 신뢰성을 유지함과 동시에 자료 처리율을 최대화할 수 있는 버스 제어기 구성, 즉 T_{BC}와 N_{BC}를 찾고자 한다. 우리는 이와 관련한 연구를 수행한 바 있지만[7], 최적화 대상 모델을 보완하고 최적화 기법을 보다 면밀하게 검토함으로써 확장 연구를 수행하였다.

3. 샘플링 기반 최적화 기법을 이용한 BC 구성

샘플링 기반 최적화 기법은 일반적으로 다음과 같이 수행된다. 먼저, 대상 시스템에서 최적화 결과가 나타날 수 있는 전체 공간으로부터 임의로 샘플을 추출한다. 그리고 추출된 샘플을 시스템 목적함수 또는 비용함수에 대입시켜 평가한다. 적절한 평가 결과를 갖는 샘플이 나올 때까지 위의 과정을 반복하다가 수렴조건에 도달하게 되면 반복을 멈추고 최종 결과로부터 최적해를 획득하게 된다. 본 장에서는 먼저 시스템 제약조건들을 유도한다. 그 다음에, 대표적인 샘플링 기반 최적화 기법들인, MCO(Monte-Carlo Optimization) 기법과 PSO(Particle Swarm Optimization) 기법을 적용하는 방법을 기술한다.

3.1 시스템 제약조건

먼저 원격 단말기에서 수신한 MMC에 대한 응답 메시지를 응답 버퍼에 쓰기 위한 시간(T_{RW})부터 분석한다. 응답 버퍼에 쓰는 태스크보다 MMC 수신과 관련된 태스크의 우선순위가 높기 때문에, T_{RW}는 MMC 수신 측면에서 나누어서 생각해야 한다. MMC를 수신하는 중에 하나의 응답 메시지를 응답 버퍼에 쓰는데 걸리는 평균 시간을 t_{RW1}라고 하고, 수신이 종료된 후에 하나의 응답 메시지를 응답 버퍼에 쓰는데 걸리는 평균 시간을 t_{RW2}라고 하자. 그러면 MMC를 구성하는 N_{BC}개의 메시지 중에서, MMC 수신이 완료되기 전까지 처리할 수 있는 메시지의 개수는 다음과 같다.

$$N_{RW1} = \left\lceil \frac{t_{MG}(N_{BC}-1)}{t_{PD}+t_{RW1}N_{RD}} \right\rceil \tag{1}$$

여기서 N_{RD}는 수신한 메시지 하나에 대한 응답 메시지의 개수이고, ⌈ ⌉은 실링(ceiling) 함수를 나타낸다. 그러면 T_{RW}는 다음과 같이 계산될 수 있다.

$$T_{RW} = N_{RW1}t_{RW1}N_{RD} + (N_{BC}-N_{RW1})t_{RW2}N_{RD} \tag{2}$$

MMC를 구성하는 메시지 하나를 처리하기 위해 요구되는 평균시간(t_{PD})과 ISR 처리시간(t_{PR})을 알 때, 원격 단말기에서 요구되는 총 시간은 다음과 같이 계산될 수 있다.

$$T_{RT} = N_{BC}(2t_{PR}+t_{PD}) + 2T_{PH} + T_{RW} \tag{3}$$

결국, 첫 번째 시스템 제약조건은 자료 손실 방지 측면에서 다음과 같이 설정된다.

$$2T_{BC} > T_{RT} \tag{4}$$

단, 이 제약조건을 만족시키기 위해 과도하게 T_{BC}를 증가시키는 것은 자료 처리율의 저하를 유발한다.

따라서 자료 처리율의 극대화를 위한 추가적인 시스템 제약조건이 필요하다. 일반적으로 경험적 방법에 의해 버스 제어기를 설계할 때, 실시간 시스템에서 예상치 못한 상황에 대비하여 설계 여유를 두게 된다. 하지만 본 논문과 같이 분석적인 접근을 할 경우, 이 여유를 최소화할 수 있다. 즉, 전체 T_{BC}에서 MMC 전송에만 사용되는 시간을 제외한 여유 시간을 최소화하기 위한 제약조건을 설정한다. 자료 손실이 없으면서 설계 여유를 최소화할수록 자료 처리율은 높아진다는 특성을 반영하여 두 번째 제약조건을 다음과 같이 설정한다.

$$N_{BC}t_{MG} \le (1-\alpha)T_{BC} \tag{5}$$

여기서 $0 \le \alpha < 1$이다. α가 커질수록 설계 여유는 커지게 된다.

샘플링 기반 최적화 기법 적용을 위해 위에서 설정된 두 제약조건들을 이용하여 목적함수를 설계한다. 실제 시스템에 존재할 수 있는 불확실성을 고려하여, T_{BC}와 N_{BC}를 나타내는 n번째 샘플, $[x_n, y_n]$에 대해 수식 (4)와 수식 (5)를 다음과 같이 다시 쓴다.

$$\Lambda_n = 2y_n - \left[x_n\left(2\widehat{t_{PR}}+\widehat{t_{PD}}\right)+2T_{PH}+T_{RW}\right] \tag{6}$$

$$\Phi_n = (1-\alpha)y_n - x_n t_{MG} \tag{7}$$

여기서 $\widehat{t_{PR}}$과 $\widehat{t_{PD}}$는 확률분포의 Three-Sigma Rule에 따라 각각 $\widehat{t_{PR}} = \overline{t_{PR}} + 3\sigma_{PR}$ 이고, $\widehat{t_{PD}} = \overline{t_{PD}} + 3\sigma_{PD}$ 이다. $\overline{t_{PR}}$와 σ_{PR}는 t_{PR}의 평균값과 표준편차이다. $\overline{t_{PD}}$와 σ_{PD}는 t_{PD}의 평균값과 표준편차이다. Λ_n가 증가할수록 자료 손실 확률이 낮아지게 되고, Φ_n가 감소할수록 자료 처리율은 높아지게 된다.

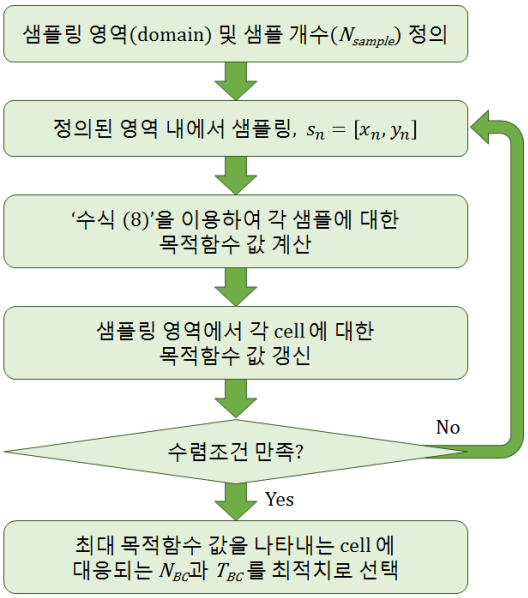


Fig. 3. Monte-Carlo Optimization

3.2 Monte-Carlo Optimization(MCO)

MCO는 최적화 대상의 확률적 분포를 반복 가능한 실험 또는 시뮬레이션을 통해 최적치를 찾는 기법이다. 최적화 제약조건들이 비선형성을 가질 경우, 복잡한 변화율 계산 없이 최적치를 찾을 수 있는 샘플링 기반 기법이다. 본 연구에 적용된 MCO의 전체적인 수행 절차는 Fig. 3과 같다. 먼저 샘플 개수(N_{sample})를 설정하고, 샘플링 영역(domain)은 가능한 N_{BC} 및 T_{BC} 범위로 설정한다. 그리고 본 시스템의 특성을 고려하여 균일 분포(uniform distribution)를 따라 샘플링을 수행한다.

n번째 샘플 $s_n = [x_n, y_n]$은 $0 < x_n \le N_{BC,\max}$ 및 $0 < y_n \le T_{BC,\max}$에서 무작위로 추출된다. 그리고 시스템의 특성을 고려하여 앞에서 유도된 수식 (6)과 수식 (7)을 이용하여, MCO를 위한 목적함수를 다음과 같이 설계하였다.

$$\Psi_n = \frac{w\Lambda_n}{(1-w)\Phi_n} \tag{8}$$

여기서 w는 가중치이고, 본 연구에서는 0.5로 설정되었다. 목적함수의 값이 높아질수록 모델의 최적값에 가까워지게 된다. 계산된 목적함수의 값을 이용하여 대응되는 샘플링 영역의 셀(cell)을 갱신한다.

MCO의 수렴조건은 미리 설정한 최대 샘플링 횟수에 도달하는 것이다. 수렴조건을 만족하면 최종적으로 샘플링 영역의 모든 셀들 중 최대 목적함수 값을 나타내는 셀에 대응되는 N_{BC}와 T_{BC}를 최적값으로 선택한다.

3.3 Particle Swarm Optimization(PSO)

PSO는 군집 지능 관점에서 개발된 샘플링 기반 최적화 기법이다[8]. 설정된 영역으로부터 샘플링을 하는 과정은 MCO와 유사하지만, 추출된 샘플을 샘플링 영역 내에서 반복적으로 이동시키며 보다 적극적으로 최적 지점을 찾아 간다는 점에서 MCO와 다르다. PSO는 비교적 구현이 간단하면서도 좋은 성능을 보여 왔기 때문에, 다양한 분야에 적용되어 왔다.

PSO에서 추출된 샘플(입자)을 이동시키는 원리는 다음과 같다. 첫 번째, 각 입자는 최적화 대상(target)에 이끌린다. 두 번째, 각 입자는 이동 경로를 기억하기 위한 메모리를 갖고 있다. 세 번째, 모든 입자들은 속도와 위치 정보를 공유한다. 마지막으로, 각 입자는 반복적인 이동을 통해 최적화 대상으로 수렴한다. 본 연구에 적용된 PSO의 전체적인 수행 절차는 Fig. 4와 같다.

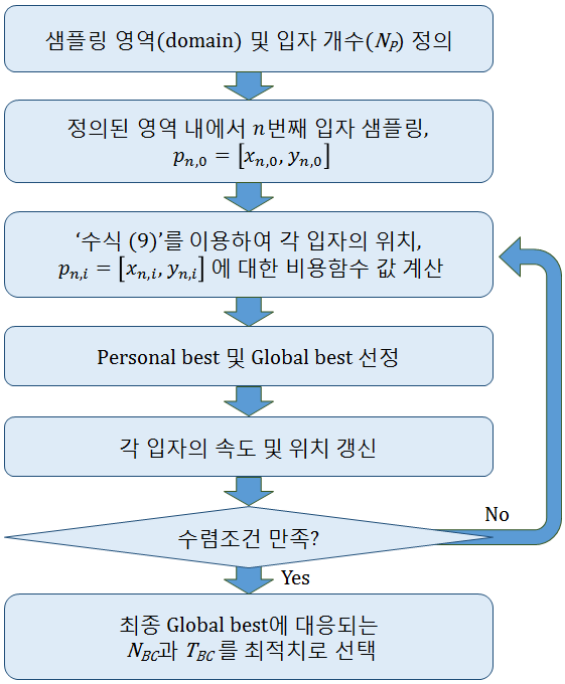


Fig. 4. Particle Swarm Optimization(PSO)

먼저 입자의 개수(N_P)를 설정하고, 샘플링 영역은 가능한 N_{BC} 및 T_{BC} 범위로 설정한다. 그리고 본 시스템의 특성을 고려하여 균일 분포(uniform distribution)를 따라 샘플링을 수행한다. 이 때, n번째 입자의 초기 위치, $p_{n,0}=[x_{n,0}, y_{n,0}]$는 $0<x_{n,0}\le N_{BC,\max}$ 및 $0<y_{n,0}\le T_{BC,\max}$에서 무작위로 추출된다.

PSO를 위한 비용함수는 앞에서 유도된 수식 (6)과 수식 (7)을 이용하여 n번째 입자의 i번째 반복에 대해 다음과 같이 설계하였다.

$$Z_{n,i}=\frac{1}{w\Lambda_{n,i}}+(1-w)\Phi_{n,i} \tag{9}$$

여기서 w는 가중치이다.

i번째 반복(iteration)에서, n번째 입자 $p_{n,i}$는 자신의 전체 이동 경로(history)에서 비용함수 $Z_{n,i}$를 최소화하는 위치를 개별최적위치(personal best) $p_{n,pb}$로 선정한다. 또한 i번째 반복에서, 전체 입자들 중 비용함수 $Z_{n,i}$를 최소화하는 위치를 전역최적위치(global best) p_{gb}로 선정한다.

이제 i번째 반복에서 선정된 개별최적위치들과 전역최적위치를 이용하여, n번째 입자의 속도($v_{n,i}$)와 위치($p_{n,i}$)를 다음과 같이 갱신한다.

$$v_{n,i+1}=\kappa\left[v_{n,i+1}+\varepsilon_1(p_{gb}-p_{n,i})+\varepsilon_2(p_{n,pb}-p_{n,i})\right] \tag{10}$$

$$p_{n,i+1}=p_{n,i}+v_{n,i+1} \tag{11}$$

여기서 κ은 입자의 속도를 제한하는 긴축(constriction) 계수 역할을 하고 1보다 작은 값으로 설정된다. ε_1와 ε_2은 각각 입자가 p_{gb}와 $p_{n,pb}$로 이끌리는 세기를 나타내고 균일 분포를 따라 무작위로 추출되는 1보다 작은 값이다.

초기에 균일하게 흩어져있던 입자들은 반복이 진행됨에 따라 특정 지점으로 수렴하게 된다. PSO의 수렴조건은 두 가지이다. 하나는 사전에 설정된 최대 반복 횟수에 도달하는 것이다. 다른 하나는 전역 최적치의 변화량이 설정된 값보다 작아졌을 때이다. 최종적으로 수렴조건을 만족하게 되면, 최종 전역최적위치에 대응되는 N_{BC} 및 T_{BC}를 시스템의 최적값으로 선정한다.

Table 1. MIL-STD-1553B system parameters

시스템 변수	설정값(μs)
ISR 처리시간 평균 ($\overline{t_{PR}}$)	38.30
ISR 처리시간 표준편차 (σ_{PR})	7.86
수신메시지 처리시간 평균 ($\overline{t_{PD}}$)	139.19
수신메시지 처리시간 표준편차 (σ_{PD})	60.92
수신 중 응답메시지 쓰기시간 (t_{RW1})	100
수신 후 응답메시지 쓰기시간 (t_{RW2})	10
상위 우선순위 태스크 처리시간 (T_{PH})	300

4. 실험 결과

본 연구에서 제안된 방법의 성능 평가는 다음과 같이 수행되었다. 먼저 제안된 방법을 별도의 시뮬레이션 환경에서 반복적으로 수행했다. 이 과정에서 제안된 방법에 사용된 각종 변수들에 따른 성능을 자세하게 평가하고 검증했다. 그리고 최종적으로 획득된 최적 N_{BC} 및 T_{BC}를 Fig. 1과 같은 실제 MIL-STD-1553B 통신 시스템에 적용하였다. 이 과정에서 제안된 방법에 의해 획득된 N_{BC} 및 T_{BC}가 아닌 버스 제어기(BC) 구성값들도 적용하고 자료 처리율을 비교함으로써, 제안된 방법의 성능을 검증하였다.

샘플링 영역 설정을 위한 $N_{BC,\max}$ 및 $T_{BC,\max}$는 시스템 요구사항을 반영하여 각각 13개와 20ms로 설정되었다. 시스템 제약조건 계산을 위한 시스템 변수들은 실험적으로 측정된 값을 바탕으로 설정되었고, 그 값은 Table 1과 같다.

4.1 MCO 기반 버스 제어기 구성 최적화 결과

샘플의 개수를 500개로 설정하고 설계 여유(α)를 0으로 설정한 후, MCO를 수행한 결과는 Fig. 5와 같다. 최대 목적함수를 나타내는 버스 제어기 구성, N_{BC} 및 T_{BC}는 각각 10개와 8 ms였고, 최대 자료 처리율은 1.25 msg/ms였다. MCO에 의해 획득된 버스 제어기 구성이 실제 최적값인지 검증하기 위해, 전체 샘플링 영역에 대한 전수 조사를 수행하였고 동일한 결과를 확인할 수 있었다. 따라서 본 연구에 적용된 MCO가 성공적으로 수행되었음을 알 수 있다.

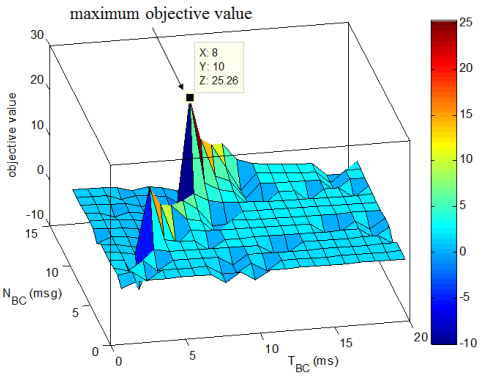


Fig. 5. The graphical result of the MCO

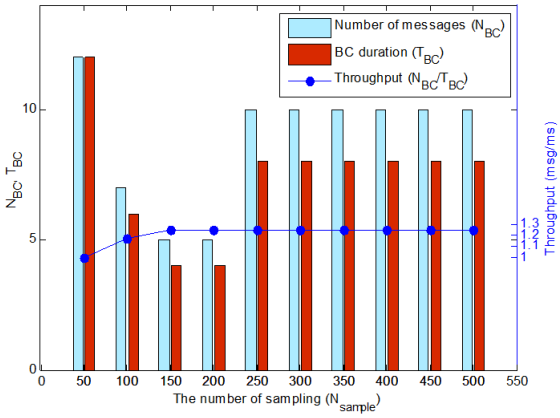


Fig. 6. The results of the MCO according to the number of samples

MCO의 성능을 보다 자세히 검증하기 위해, 샘플의 개수를 변화시켜 가면서 MCO를 수행하였고, 그 결과는 Fig. 6과 같다. 샘플 개수가 50개 및 100개일 때는 MCO가 최적값을 찾지 못함에 따라 자료 처리율이 각각 1 msg/ms와 1.17 msg/ms를 나타냄으로써 최적 자료 처리율인 1.25 msg/ms에 미치지 못했다. 하지만, 샘플 개수가 150개 이상일 때에는 최적 자료 처리율인 1.25 msg/ms을 나타냄으로써 버스 제어기 구성의 최적값을 성공적으로 획득했음을 알 수 있다.

4.2 PSO 기반 버스 제어기 구성 최적화 결과

입자의 개수를 300개, 설계 여유(α)를 0, 긴축계수(κ)를 0.25로 설정한 후, PSO를 수행한 결과는 Fig. 7과 같다. 녹색 작은 점들은 설정된 샘플링 영역에서 균일 분포를 따라 무작위로 추출된 입자들의 위치를 나타낸다. 검은색 원들은 PSO가 진행 중인 상황에서 입자들의 중간 위치를 나타낸다. 파란색 큰 점들은 수렴조건 만족 후 PSO가 종료되었을 때의 입자들의 최종 위치를 나타내고, 전역최적위치(global best)는 붉은색 가위표로 표시되었다.

이 결과를 통해 입자들이 특정 지점으로 수렴했음을 알 수 있다. 본 연구에서는 시스템 특성을 고려하여 N_{BC} 및 T_{BC}를 정수로 가정했기 때문에, 최종 전역최적위치를 반올림한 값을 최적치로 사용했다. 즉, PSO 수행 결과 N_{BC} 및 T_{BC}는 MCO 수행 결과 및 전수 조사 결과와 마찬가지로 각각 10개와 8 ms였고, 최대 자료 처리율은 1.25 msg/ms였다. 따라서 본 연구에 적용된 PSO가 성공적으로 수행되었음을 알 수 있다.

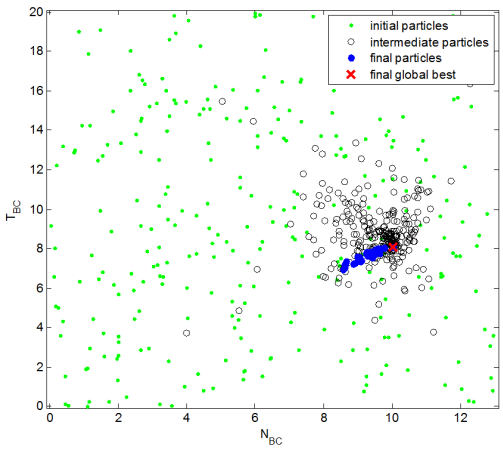


Fig. 7. The graphical result of the PSO

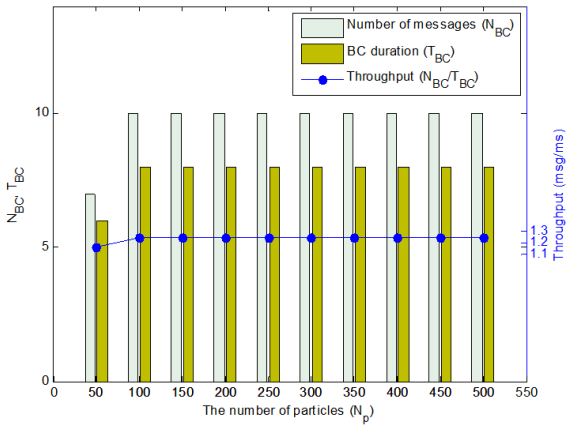


Fig. 8. The results of the PSO according to the number of particles

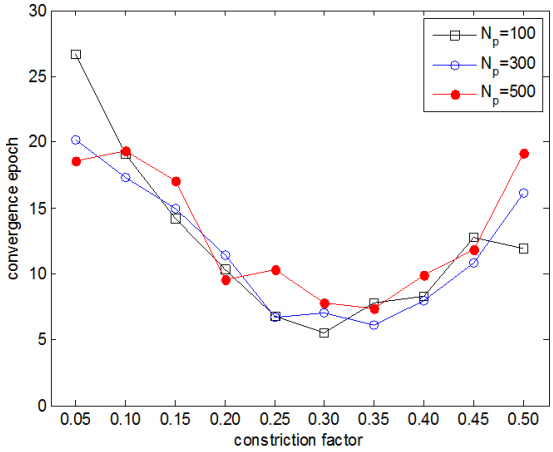


Fig. 9. The convergence results of the PSO according to the constriction factor(κ)

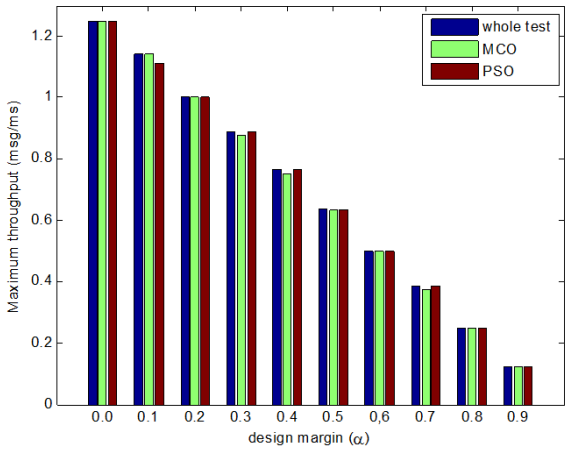


Fig. 10. Maximum throughputs with optimization results according to the design margin(α)

PSO의 성능을 보다 자세히 검증하기 위해, 입자의 개수를 변화시켜 가면서 PSO를 수행하였고, 그 결과는 Fig. 8과 같다. 입자의 개수가 50개일 때는 자료 처리율이 1.17 msg/ms로 나타냄으로써 최적 자료 처리율인 1.25 msg/ms에 미치지 못했다. 하지만, 입자의 개수가 100개 이상일 때에는 지속적으로 자료 처리율이 항상 1.25 msg/ms을 나타냄으로써 버스 제어기 구성의 최적값을 성공적으로 획득했음을 알 수 있다.

더 면밀한 평가를 위해, 입자의 개수 및 입자의 이동 속도를 제어하는 긴축 계수(κ)에 따른 PSO의 수렴 속도를 Fig. 9와 같이 나타내었다. Fig. 9에서 수렴 속도는 수렴에 필요한 반복 횟수(convergence epoch)로 표현하였고, 그 값이 낮을수록 수렴 속도는 빠른 것이다. 각 결과는 동일 조건에서 10회 반복 실행한 결과들의 평균값이다. 입자의 개수는 수렴 속도에 크게 영향을 주지 않았다. 긴축 계수가 높을수록 입자의 속도를 적게 제한하기 때문에 높을수록 수렴 속도가 빨라질 것으로 예상되었으나, 특정 값을 넘게 되면 오히려 수렴 속도가 느려짐을 확인할 수 있었다. 따라서 PSO를 적용할 때에는 적절한 긴축 계수를 설정하는 것이 중요함을 알 수 있었다. 본 시스템 모델에는 0.2~0.4의 값이 적절함을 확인할 수 있다.

4.3 시스템 요구사항 변경에 따른 결과

앞의 결과들은 모두 시스템 요구사항 중 설계 여유(α)가 0인 경우의 결과들이다. 설계 여유의 변화에 따른 적용 결과를 Fig. 10과 같이 나타내었다.

MCO는 300회의 샘플링을 수행하였고, PSO는 300개의 입자를 사용하였다. 전반적으로 MCO와 PSO 모두 전수 조사 결과와 거의 동일한 결과를 나타내었으나, MCO의 경우 설계 여유가 0.3, 0.4, 0.7인 경우에는 전수 조사 결과와 조금 다른 결과를 나타내었고, PSO의 경우 설계 여유가 0.1인 경우에는 전수 조사 결과와 조금 다른 결과를 나타내었다. 두 기법들 간 엄밀한 성능 비교는 쉽지 않지만, 동일한 개수의 샘플 및 입자를 사용하는 경우, 일반적으로 PSO가 조금 더 높은 정확도를 나타내었다. 반면에, 기법 구현 복잡도 및 수행시간은 MCO가 더 낮았다. 따라서 대상 시스템의 목적에 부합하는 기법을 선택하는 것이 좋다.

4.4 실제 시스템 적용 결과

앞에서 획득한 버스 제어기의 최적 구성, 즉 N_{BC} 및 T_{BC}를 각각 10 msg와 8 ms로 실제 MIL-STD-1553B 통신 시스템의 버스 제어기에 적용하였다. 또한 성능 검증을 위해 다른 버스 제어기 구성도 적용해보았다. 본 연구에서 사용된 MIL-STD-1553B 통신 시스템에서, 버스 제어기는 Windows의 실시간 확장 패키지를 적용한 워크스테이션이 사용되었고, 총 6632개의 메시지로 구성된 임무 자료를 N_{BC}개 메시지로 구성된 MMC 단위로 T_{BC}마다 원격 단말기 역할을 하는 유도제어장치로 전송한다. 유도제어장치는 VxWorks 기반 프로그램을 이용하여 임무 자료를 실시간으로 수신 및 처리한다. 버스 라인 상의 모든 자료는 별도의 모니터링 컴퓨터를 이용하여 관찰하면서 저장하였다.

Table 2. Experimental results with the real MIL-STD-1553B communication system

N_{BC}	T_{BC}	전송 시간
8 msg	6 ms	Fail
8 msg	7 ms	6.516 s
8 msg	8 ms	7.447 s
8 msg	9 ms	9.217 s
8 msg	10 ms	11.384 s
9 msg	6 ms	Fail
9 msg	7 ms	Fail
9 msg	8 ms	6.333 s
9 msg	9 ms	7.920 s
9 msg	10 ms	9.828 s
10 msg	6 ms	Fail
10 msg	7 ms	Fail
10 msg	8 ms	5.374 s
10 msg	9 ms	6.819 s
10 msg	10 ms	8.520 s

다양한 버스 제어기 구성을 실제 임무 자료 전송 시스템에 적용해보았고, 임무 자료 전송 시간을 Table 2와 같이 측정하였다. 버스 제어기에서 전송한 임무 자료와 유도제어장치에 수신된 자료가 다른 경우는 '전송실패(Fail)'로 처리하였다. 이는 버스 제어기가 과도하게 빠른 주기로 임무 자료를 전송함에 따라 자료가 있는 유도제어장치의 버퍼에 덮어씀으로 인해 자료가 손실되었기 때문이다. 전송실패 없이 가장 효율적으로 임무 자료를 전송한 버스 제어기 구성은 N_{BC} 및 T_{BC}가 각각 10 msg와 8 ms일 때였다. 이는 본 논문에서 제안된 방법에서 획득한 결과와 동일하다. 따라서 제안된 방법은 성공적으로 동작했음이 검증되었다.

5. 결 론

본 논문은 다중메시지체인 및 더블버퍼링 기술이 적용된 MIL-STD-1553B 통신에서 효율적 임무 자료 전송을 위한 샘플링 기반 최적화 기법 적용 방법을 제안하였다. 제안된 방법을 통해 버스 제어기를 구성하는 최적 메시지 개수와 최적 전송 주기를 획득할 수 있었다. 제안된 방법의 성능은 다양한 변수 및 시스템 요구사항에 따라 면밀하게 분석 및 평가되었다. 또한, 제안된 방법에 의해 획득된 결과가 실제 실시간 시스템에서도 최적 자료 처리율을 보장함을 통해 그 성능이 검증되었다.

References

[1] MIL-STD-1553B NOTICE II, Department of Defense, Washington D. C. 20360, Sep. 1986.

[2] MIL-STD-1553B Designer's Guide, Data Device Corporation, 1998.

[3] MIL-STD-1553B Evolves with the Times, White Paper, Data Device Corporation, 2010.

[4] J. Zhang, M. Liu, G. Shi, and W. Pan, "A MIL-STD-1553B Bus Command Optimization Algorithm based on Load Balance," Applied Mechanics and Materials, Vol. 130-134, pp. 3839-3842, 2012.

[5] K. Padmanabham, P. Kanugo, K. N. Raju, and M. Chandrashekar, "A Review of MIL-STD-1553 Bus Trends and Future," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, No. 8, pp. 119-122, 2016.

[6] K. P. Kim, K. H. Ahn, Y. S. Kwon, S. J. Yun, and S. H. Lee, "Analysis and Implementation of High Speed Data Processing Technology using Multi-Message Chain and Double Buffering Method with MIL-STD-1553B," Journal of the Korea Institute of Military Science and Technology, Vol. 16, No. 4, pp. 422-429, 2013.

[7] H. C. Lee, K. P. Kim, and Y. S. Kwon, "PSO-based Optimization of the BC Message Chain Structure for Multi-Message Chain and Double Buffering in MIL-STD-1553B," KIMST Annual Conference Proceedings, pp. 1176-1177, June, 2015.

[8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proceedings of IEEE International Conference on Neural Networks, pp. 1942-1948, WA, Nov. 1995.