http://dx.doi.org/10.5515/KJKIEES.2012.23.2.196

Simulated Annealing을 이용한 추계적 레이더 빔 스케줄링 알고리즘

Stochastic Radar Beam Scheduling Using Simulated Annealing

노지은 · 안창수 · 김선주 · 장대성* · 최한림*

Ji-Eun Roh · Chang-Soo Ahn · Seon-Joo Kim · Dae-Sung Jang* · Han-Lim Choi*

요 약

능동 위상 배열 레이더(AESA Radar: Active Electronically Scanned Array radar)는 전자적으로 빔을 조향함으로써 빔 조향 시간이 비약적으로 빨라져 기존의 기계식 빔 조향 레이더에 비해 레이더에서 수행할 수 있는 다중 임무 처리 능력이 크게 향상되었다. 이러한 이유로 레이더에 주어진 시간, 에너지, 처리 능력 등의 한정된 자원을 실시간으로 효율적으로 관리, 운용할 수 있는 레이더 자원 관리 기술의 중요성이 크게 대두되었다. 그 중 레이더 빔 스케줄링 기술은 레이더 자원 관리의 핵심적인 요소라 할 수 있다. 본 논문에서는 simulated annealing을 이용한 추계적 레이더 빔 스케줄링 알고리즘을 제안하고, 이를 기존의 dispatching rule에 기반한 빔 스케줄링 기법과 비교하였다. 빔 처리 지연도(latency)와 주어진 시간 내에서 처리할 수 있는 빔의 개수 측면에서 스케줄링 결과를 비교하여 성능의 우월성을 입증하였으며, 또한 실시간성을 보장하면서도 기존의 규칙 기반 알고리즘보다 성능이 우수함을 보였다.

Abstract

AESA radar is able to instantaneously and adaptively position and control the beam, and such adaptive beam pointing of AESA radar enables to remarkably improve the multi-mission capability, compared with mechanically scanned array radar. AESA radar brings a new challenges, radar resource management(RRM), which is a technique efficiently allocating finite resources, such as energy and time to each task in an optimal and intelligent way. Especially radar beam scheduling is the most critical component for the success of RRM. In this paper, we proposed stochastic radar beam scheduling algorithm using simulated annealing(SA), and evaluated the performance on the multi-function radar scenario. As a result, we showed that our proposed algorithm is superior to previous dispatching rule based scheduling algorithm from the viewpoint of beam processing latency and the number of scheduled beams, with real time capability.

Key words : AESA Radar, Radar Resource Management, Stochastic Radar Beam Scheduling, Simulated Annealing

Ⅰ. 서 론

3차원 다기능 레이더(MFR: Multi-Function Radar)는 실시간으로 다수의 표적을 탐지, 추적하여 거리 및 각도 정보를 제공하는 최신 레이더 시스템이다. 표적의 탐지로부터 추적까지를 실시간으로 처리하기 위해서는 빔 조향이 용이한 위상 배열 안테나 기술과 다양한 실시간 레이더 신호처리 기술과 더불어, 레이더의 한정된 자원을 효과적으로 활용하여 레이더 전체 성능을 향상시키는 레이더 자원 관리(RRM:

국방과학연구소(Agency for Defense Development)
*한국과학기술원(KAIST)
· 논 문 번 호 : 20111020-131
· 교 신 저 자 : 노지은(e-mail : jeroh@add.re.kr)
· 심사일자 : 2011년 12월 16일 · 수정완료일자 : 2011년 12월 22일

Radar Resource Management) 기술의 중요성이 크게 대두되었다. 특히, 능동 위상 배열 레이더(AESA Radar: Active Electronically Scanned Array radar)는 전자적으로 빔을 조향함으로써 빔 조향 시간이 비약적으로 빨라, 기존의 기계식 빔 조향 레이더에 비해 레이더에서 수행할 수 있는 다중 임무 처리 능력이 크게 향상되었다. 이러한 다중 임무 처리 능력을 극대화하기 위해서는 레이더에 주어진 시간, 에너지, 처리 능력 등의 한정된 자원을 실시간으로 효율적으로 관리, 운용할 수 있는 레이더 자원 관리 기술이 필요하다.

레이더의 한정된 자원을 효율적으로 사용하기 위해 적응적 파형 선택, 표적 위협도 분석을 통한 표적별 차등 처리, 적응적 추적 주기 기법 등이 연구되고 있으며, 그 중 레이더 빔 스케줄링 기술은 레이더 자원 관리의 핵심적인 요소라 할 수 있다. 레이더 빔 스케줄링은 레이더 임무들의 수행 시간과 순서를 임무의 우선 순위, 전술적 요구 조건 및 임무 처리 성능 지표에 입각하여 실시간으로 결정하는 기술이다.

본 논문에서는 이러한 레이더 빔 스케줄링을 위해 Simulated Annealing(SA)을 이용한 추계적(Stochastic) 빔 스케줄링 알고리즘을 제안하였고, 이를 기존의 dispatching rule에 기반한 스케줄링 알고리즘과 비교하여 그 성능을 분석하였다. 빔의 처리 시간 지연도(latency), 주어진 시간 내에 스케줄링된 전체 빔의 개수, frame time 측면에서 성능을 비교해 보았는데, SA를 이용한 추계적 빔 스케줄링 기법이 기존의 방법보다 성능이 우월하며, 동시에 실시간성을 보장할 수 있음을 입증하였다. 본 논문의 Ⅱ장에서는 다기능 레이더 시스템을 위한 레이더 자원 관리 연구 중, 빔 스케줄링에 관련된 기존 연구에 대해 설명하고, Ⅲ장에서는 SA를 이용한 추계적 빔 스케줄링 알고리즘을 제안하였다. Ⅵ장에서는 시뮬레이션을 통해 그 결과를 분석하고, 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. 레이더 빔 스케줄링 기존 연구

레이더 빔 스케줄링은 우선 순위 할당, 스케줄링 방법의 유연성, 최적화 가능성에 따라 비적응형과 적응형 크게 두 가지 형태로 구분할 수 있는데, 참고문헌 [1]의 조사에 따르면, 비적응형 기법은 임무 우선 순위가 미리 정해져 있으며, 스케줄러는 이러한 우선 순위와 정해진 휴리스틱 규칙에 따라 자원 관리를 실행하게 된다. 적응형 기법의 경우 좀 더 복잡한 과정을 이용해 우선 순위를 결정하거나 최적화된 해를 얻기 위하여 스케줄링 단계에서 다양한 기법들을 활용하는 것이 특징이다. 문헌에서 주로 언급되고 있는 비적응형 기법으로는 Butler형[2]과 Orman형[3]이 있다. 두 기법 모두 빔의 고정된 우선 순위가 정해져 있으며, 스케줄링은 시간축(time-line)을 따라 진행하면서 순차적인 처리 과정을 통해 결정하도록 되어 있다.

Butler형은 요청된 임무들에 타임 밸런스를 배정하여 순차적으로 비교하도록 구성되어 있다. 타임 밸런스는 스케줄링마다 증가하는 값으로 임무에 대해 남은 한계 시간을 나타낸다. 즉, 음수라면 한계 시각까지 남은 시간을 나타내고, 양수라면 이미 한계 시각을 지나쳐 지연되고 있다는 의미이다. 만약 임무가 정확히 요청된 시간 안에 처리된다면 타임 밸런스는 0이 된다. 타임 밸런스와 주어진 임무 우선 순위에 따라 순차적인 처리 과정을 반복하게 되는데, 우선 순위에 입각하여 일차적으로 임무를 선택하고, 그중에 양인 타임 밸런스를 가지는 임무를 선택한 후 모든 타임 밸런스를 증가시켜 시간축에서 양의 방향으로 이동하여 계속 스케줄링을 진행한다. 이 방법의 단점은 우선 순위가 높은 임무(추적)가 지속적으로 많이 요청되는 경우, 우선 순위가 낮은 임무(탐색)가 계속 뒤로 밀려서 수행되지 않을 수 있다는 것이다. 대신에 시간축 상에 빈 공간을 남겨 두지 않으므로, 한정된 시간 자원을 모두 사용한다는 이점이 있다.

Orman형은 일정 시간 간격으로 떨어진 두 개의 다른 임무의 결합(coupled-task)을 하나의 임무로 생각한다. 하나의 임무는 레이더 펄스를 보내는 시간, 대기 시간, 반송 펄스를 받는 시간으로 구성되고, 성능 개선을 위해 대기 시간을 활용할 수도 있다. 대기 시간의 길이는 레이더가 수행하는 임무의 특성과 표적의 거리, 탐색 영역, 송신 시간 등에 따라 좌우된다. Orman형 역시 Butler형과 마찬가지로 우선 순위가 높은 임무들에 의해 낮은 우선 순위를 가지는 임무의 처리가 지연될 수 있다. 이는 스케줄러가 고정된 우선 순위를 사용하고, 가장 높은 우선 순위를 가

지는 임무를 항상 먼저 처리하는 특성 때문이다. 참고문헌 [4]는 Orman형 스케줄러를 사용하면 Butler의 결과에 비해서 임무처리 요청 시각 가까이에 임무들을 스케줄링할 수 있음을 지적하고 있지만, Orman형 스케줄러는 레이더의 가용 시간을 모두 사용하지 못한다는 큰 단점이 있다.

기존의 휴리스틱을 이용한 규칙 기반 비적응형 기법은 구현이 용이하고 빠르며, 비교적 짧은 시간 안에 제한된 문제에 한해 적절한 해를 찾을 수 있다는 장점이 있다. 하지만 일반적으로 성능이 최적화됨을 보장할 수 없다는 단점이 있다.

이러한 한계를 극복하기 위해 소프트 컴퓨팅(ANN, Fuzzy)이나 전문가 시스템을 사용한 인공 지능 기반 알고리즘[5],[6], 은닉 마르코프 모델(HMM)이나 마르코프 결정 모델(MDP)구조 하에서 동적 프로그래밍을 통해 문제를 해결하는 방법[7], QoS(Quality of Service)를 목적 함수로 두는 Q-RAM 알고리즘[8]~[10] 등이 제안되었으며, 이러한 알고리즘은 빔 스케줄링 문제를 파형 최적화, 추적 주기 최적화 등과 결합되어 해결하고자 한다. 이러한 최적화에 기반한 적응형 기법은 학계에서는 활발히 연구 중이지만, 실제 레이더 시스템에 적용하여 실시간으로 운용되기에는 알고리즘의 복잡도나 신뢰성 측면에서 아직은 기술이 성숙되어 있지 않다고 볼 수 있다.

참고문헌 [11]에서는 유전자 알고리즘(Genetic Algorithm, GA)을 이용한 위성 임무 스케줄링 최적화 기법을 제안했는데, 업무가 실행할 순서대로 정렬되었다는 가정 하에, 유전자 알고리즘의 목적 함수를 최대화하는 업무들을 선택하여 최종적으로 수행할 업무 리스트를 결정하는 방식이다. 위성 업무라서 업무 마감시간(deadline)이 일(day) 단위이고, 유전자 알고리즘을 적용하여 스케줄링까지 걸리는 시간이 분(minute) 단위이므로 위성 임무 스케줄링에서는 이러한 방법이 가능하나, 훨씬 더 짧은 시간에서 스케줄링이 이루어져야 하는 레이더 시스템에서는 동일한 방법으로 실시간성을 만족할 수 없다. 그 외에 Job Sop, Flow Shop 등의 문제를 다루는 실시간 시스템에서 GA, SA, Tabu search와 같은 전역 및 국부 탐색을 이용한 최적화 방법론이 소개되긴 했지만[12],[13] 레이더 빔 스케줄링에 이러한 추계적인 메타 휴리스틱 방법론을 적용한 기존 연구는 없다.

정리하면, 본 논문은 기존의 규칙 기반 알고리즘의 성능을 향상하기 위해, 메타 휴리스틱 기반한 최적화 방법 중의 하나인 SA를 이용하여 추계적 레이더 빔 스케줄링 알고리즘을 제안하고, 이를 기존 방법 중 널리 알려지고 인용되는 Butler형 알고리즘과 비교하여 그 성능을 분석하고자 한다.

Ⅲ. Simulated Annealing을 이용한 레이더 빔 스케줄링

3-1 Simulated Annealing(SA) 소개

SA는 기본적으로 국소 탐색 방법을 개선하여 전역 최적화(global optimization) 문제에 대한 일반적인 확률적 휴리스틱(또는 메타 휴리스틱) 접근 방식이다. 최적화란 주어진 제약 조건 하에서 달성하고자 하는 목표를 이루기 위한 의사 결정 문제를 모형으로 정형화하고, 이에 대한 해를 구하기 위한 방법론으로 시스템의 최적화(최대의 이익, 최소의 비용, 고객서비스의 최대화, 최적의 선택)를 대상으로 하고 있다. SA의 명칭과 정신은 야금학(metallurgy)에서 담금질(annealing)에서 따온 것이다. 일반적으로 금속을 달군 뒤에 냉각하게 될 경우, 어떤 조건(온도, 기압)에서 금속을 냉각시키는지에 따라서 해당 금속의 성질은 상이하게 달라진다. 따라서 자기가 원하는 스타일의 금속을 얻기 위해서는, 담금질에서 냉각 스케줄이 매우 중요하다. 담금질에서 금속 온도가 높을 때에는 원소들이 활발하게 움직이기 때문에 원소들이 제자리를 찾기 쉬워진다. 그러나 온도가 낮을 때에는 원소들의 움직임이 안정되기 때문에 원소들이 움직임이 둔화되고 안정된 상태의 결정상태가 되기 때문에, 원소들의 랜덤한 이동이 어렵게 된다. SA란, 이러한 것들을 컴퓨터를 통해 흉내 내어, 조합 최적화 문제의 최적해를 얻고자 하는 방법이며, 다양한 최적화 문제에 적용되어 왔다[14]~[17]. SA의 가장 큰 특징은 좋지 못한 해의 채택도 어느 정도의 확률로 허락하는 것인데, 처음에는 나쁜 해로의 이동을 큰 확률로 주고, 점점 그 확률을 줄이게 된다. SA에서 나쁜 해에 대한 채택 여부를 결정하기 위해 온도의 개념이 필요하다. 즉, SA 알고리즘의 초기 온도에서 차츰 시간이 지나면 해당 온도를 조금씩 낮춰주게 되고, 온도가 높을 때는 나쁜 해로의 허락 확률

표 1. SA 적용을 위한 알고리즘 인자, 변수 및 cost function의 정의
Table 1. Definition of algorithm parameters, variables, and cost function for SA.

Algorithm parameters and variables
- Beam Definition : Beam(id, r, p, p_i, k) id: beam의 식별 ID r: 요청시간(request time), p: 처리 시간(processing time, i.e. dwell time), p_i: 빔의 우선 순위 레벨(beam priority level) k : 빔의 스케줄링 순서 (SA 과정을 통해 정해짐) - Priority_level p_i ∈ {1(plot confirmation), 2(track initiation), 3(active track), 4(surveillance)} //낮은 숫자가 높은 우선 순위를 가짐. - ct : current time (스케줄링이 이루어지는 시간) - Scheduling_Time_Interval = OO ms (SA를 통해 순서를 최적화 할 시간 구간) - SCS(Scheduling Candidate Set) = {beam(id, r, p, p_i, k) | r < ct + Scheduling_Time_Interval} - V = Sequence of beams belong to SCS (i.e. beam ∈ SCS)
Functions and Criteria

- Neighbor generation: select random two beams on V, and exchange their orders each other.
- Cost calculation
earliness_penalty_score = [penaltyp_i=1 penaltyp_i=2 penaltyp_i=3 penaltyp_i=4]
//차례로 p_i = 1, 2, 3, 4에 해당
lateness_penalty_score = [penaltyp_i=1 penaltyp_i=2 penaltyp_i=3 penaltyp_i=4]
//차례로 p_i = 1, 2, 3, 4에 해당
For each beam(id, r, p, p_i, k) on V schedule //k = 1, 2, 3, ⋯ |SCS|
Let r' = estimated request time of the kth beam
r' = ct + $\sum_{m=1}^{m=k-1} p_m$, p_m is processing time of mth position beam(id, r, p, p_i, m)
Let escore = max(0, r − r'), //earliness of the kth beam,
then, escore * earliness_penalty_score(i, p_i)
Let lscore = max(0, r' − r) //lateness of the kth beam,
then, lscore * lateness_penalty_score(i, p_i)
Let pscore = escore + lscore //penalty score of the kth beam
sum pscores of all beams on V schedule
- Stopping criteria : T is near zero or cost value is nearly unchangeable

이 크고, 온도가 낮을 때는 허락 확률이 낮게 된다.

3-2 SA를 이용한 레이더 빔 스케줄링 알고리즘

3-1절에서 설명한 SA의 개념을 레이더 빔 스케줄링에 적용하기 위해 먼저 표 1과 같은 알고리즘 인자, 변수, 해를 평가할 cost function 등을 정의하였다.

다음, 이를 표 2의 SA 알고리즘에 적용하였다.

SA를 적용한 빔 스케줄링 알고리즘 제안에 있어 핵심은 스케줄링 알고리즘 변수 및 cost function을 정의하는 일이라고 볼 수 있다. 먼저, Scheduling_Time_Interval은 알고리즘의 성능을 좌우하는 중요 변수중의 하나인데, 이 변수는 SA 알고리즘 적용시 매번 스케줄링이 필요한 시점에서 평균 몇 개의 빔의 순서를 한 번에 최적화하느냐를 결정한다. 한 번에 최적화해야 할 빔의 개수가 많아지면 최적해로의 수렴 속도가 늦어질 것이고, 한 번에 최적화해야할 빔의 개수가 너무 작으면 전체적으로 최적화의 효과가 떨어지게 될 것이다. 본 논문에서는 초기 해를 구하기 위해 SCS에 속한 빔들을 랜덤하게 정렬하였다. 이웃해는 현재 해에서 랜덤하게 두 개의 위치를 선택하여 두 위치에 해당하는 빔의 순서를 바꾸는 것으로 정하였다. 해당 해의 cost값은 해당 해에서 선정한 각 빔들의 처리 순서에 대해 earliness와 lateness 정도를 구하고, 이를 earliness penalty score와 lateness penalty score를 곱하여 전체를 다 더한 값으로 구하였다.

표 2. SA 스케줄링 알고리즘
Table 2. SA Scheduling algorithm.

SA_Scheduler Algorithm

Step 1. Formulation the problem parameters
Step 2. Determination of the initial schedule, generate a feasible solution V;
Step 3. Initialization the cooling parameters:
  Set the initial value of the temperature parameter, T;
  Set the temperature length L;
  Set the cooling rate F;
  Set the Boltzmann factor k;
Step 4. Select a neighbor V' of V
  Let C(V') = the cost of the schedule V'
  Compute the move value △= C(V') − C(V)
Step 5. If △<=0
  accept V' as a new solution and set V = V'
  Else if $e^{\frac{\triangle}{kT}} > rand(0, 1)$ set V = V'
  Else retain the current solution V
Step 6. Repeat Step 4 and Step 5 during L times
Step 7. Update the Temperature Tm+1 = F*Tm, m = 1, 2, 3, …
Step 8. Repeat Step 4 ~ Step 7 while stopping criteria is not met

핵심 개념을 정리하면, SA를 적용한 빔 스케줄러는 매 Scheduling_Time_Interval마다 SCS를 구하고, SCS에 속한 빔들을 대상으로 earliness_penalty_score와 lateness_penalty_score를 이용해 해의 cost 값을 가장 작게 하는 즉, penalty score를 가장 작게 하는 최적의 순서를 찾는 것이다. 이때 penalty score는 빔의 우선 순위와 레이더 시스템의 목적에 따라 적절하게 정의될 수 있다.

Ⅳ. 시뮬레이션

4-1 시뮬레이션 환경

본 논문에서 제안한 알고리즘과 기존 방법 중 널리 알려지고 인용되는 Butler형 알고리즘과 비교하기 위해 참고문헌 [2]의 5장에서 사용된 것과 동일한 시뮬레이션 환경을 구성하였다. 레이더 임무는 두 개의 탐색 영역으로 이루어진 공간을 탐색하다가 표적이 발견되면, 순차적으로 plot confirm, track initialization, active track(track update)을 수행하는 것이다. 탐색 영역의 구분에 대한 정보들은 표 3과 같다.

표적의 출현과 탐지 및 추적은 다음과 같이 가정하였다. 전체 30초의 시뮬레이션 시간에 대해 무작위로 50개의 고정된 표적을 시뮬레이션 시간의 초기 약 20초의 시간 내에 생성하고, 표적이 공간상에 최초 출현한 이후에 탐색 빔의 방향에 표적이 위치하면 plot confirm과 track initialization 및 active track을 수행하도록 하였다. 탐지에서 plot confirm까지의 시간 제약은 0.1초, track initialization은 0.1초 주기로 10회, active track을 위한 추적 갱신 주기는 0.3초로 설정하였다.

빔의 dwell time은 빔 조향 각도에 따른 SNR 손실

표 3. 시뮬레이션을 위한 탐색 영역과 관련 인자들
Table 3. Parameters of search regions for simulation.

Region	Azimuth	Elevation	Avg. dwell time(ms)	Occupancy
1	±45°	0～20°	1.42 broadside:1	20 %
2	±45°	20°～50°	3.304 broadside:2	80 %

Region	# of beams	Frame time(s)	Function time(s)	Look interval(ms)
1	283	2.010s	0.402	7.102
2	329	1.359s	1.087	4.130

을 보상하여 표적 탐지 성능이 유지될 수 있어야 한다. 레이더 빔 조향 각도에 Ψ에 대한 감쇠된 신호대 잡음비 S/N_{Ψ}는 다음과 같다[2].

$$S/N_{\Psi} = S/N_o \cos^3 \Psi$$

이에 따라 빔 조향 각도에 따른 dwell time은 위식의 역에 비례해서 증가해야 한다. 레이더의 각 탐색 영역에 대해서 function time $t_{function}$, frame time t_{frame}, look interval $\triangle t_{look}$은 탐색 영역의 레이더 시간 요구 점유율 occ_d, 총 빔의 개수 N_{beam}, 레이더 정면에서의 dwell time $t_{d,0}$에 따라 아래와 같은 식으로 주어진다.

$$t_{function} = t_{d,0} N_{beam}$$
$$t_{frame} = t_{function} / occ_d$$
$$\triangle t_{look} = t_{frame} / N_{beam}$$

3-2절에서 제안한 알고리즘을 실제 시뮬레이션에 적용하기 위해 SA 적용에 필요한 인자값들을 다음과 같이 설정하였다.

- Scheduling_Time_Interval = 10 ms
- Initial temperature parameter, T = 10;
- Temperature length L = 5;
- Cooling rate F = 0.95;
- Boltzmann factor k = 1;
- Earliness_penalty_score = [10 20 50 0]
- Lateness_penalty_score = [30 10 15 0]
- Stopping criteria : T < 0.7

앞서 언급한 것처럼 Scheduling_Time_Interval은 알고리즘의 성능을 좌우하는 중요 인자 중의 하나인데, 이 값은 SA 알고리즘 적용시 매번 스케줄링이 필요한 시점에서 평균 몇 개의 빔의 순서를 한 번에 최적화하느냐를 결정한다. 표 3에서 알 수 있듯이 시뮬레이션에서 가정한 빔의 dwell time이 1～3 ms 정도이므로 Scheduling_Time_Interval을 10 ms로 잡을 경우 한 번에 순서를 최적화해야 할 빔의 개수는 평균 5～8개 정도로 예상할 수 있다.

Penalty score에 관해서 살펴보면, 요청된 시간보다 일찍 처리됨에 따른 penalty는 active track을 가장 높이 주었는데, 그 이유는 정밀 추적을 예정된 시간보다 빨리 할 경우는 전체적으로 봤을 때 필요 이상으로 추적을 많이 하게 되어 한정된 레이더의 가용 자원을 낭비하게 되기 때문이다. 요청된 시간보다 늦게 처리되는 것에 대한 penalty는 plot confirm에 가장 높이 부과하였다. 일반적인 레이더 시스템에서 plot confirm 빔을 통해 오표적을 판단하고 오표적이 아닐 경우, confirm빔을 통해 표적의 보다 정확한 정보를 획득할 수 있게 된다. 따라서 일반적으로 탐지 후 표적이 발견된 경우 plot confirm 빔의 방사까지의 시간 제약을 강제사항으로 두고 있으며, 이러한 이유로 plot confirm이 늦어지는 경우에 대한 penalty를 가장 높이 부과하였다.

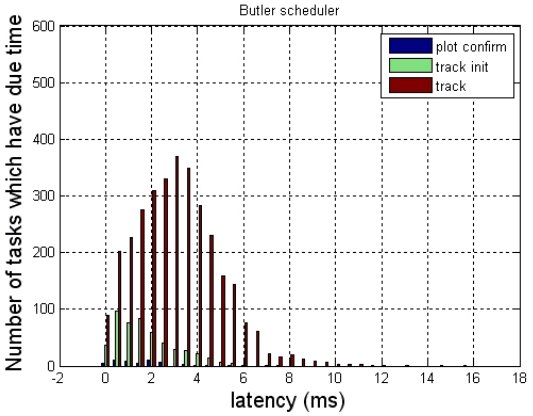
4-2 빔 스케줄링 성능 분석

4-2-1 빔 처리 지연도 비교

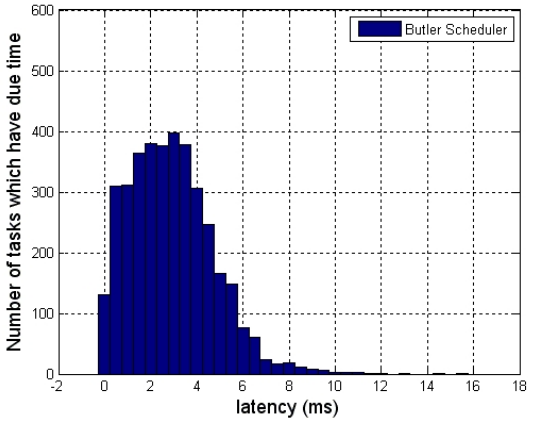
제안한 방법의 성능을 비교하기 위해 먼저 Butler 알고리즘을 적용한 결과와 SA를 적용한 스케줄링 결과에 대한 빔별 시간 지연도 및 전체 시간 지연도의 히스토그램을 분석하였다.

그림 1은 Butler 알고리즘을 적용하여 시뮬레이션한 결과이다. Butler 알고리즘은 모든 빔들이 항상 요청된 시간 이후에 처리된다는 특징이 있으며, 시뮬레이션 결과도 같은 경향을 보여준다. 지연되지 않고 요청된 제시간에 처리된 빔들은 약 130여 개 정도가 되며, 가장 분포가 높은 약 400여 개의 빔이 요청된 시간부터 3 ms 정도 지연되어 처리되고 있다. 또한 소수의 빔들이 16 ms까지 지연되기도 한다. 짧은 시간 안에 많은 수의 추적 임무들이 요청되는 상황이므로 전체 시간 지연도 결과는 정밀 추적 빔의 지연도에 크게 의존적임을 알 수 있다.

그림 2는 본 논문에서 제안한 SA를 적용한 스케줄링 결과를 보여준다. Butler 알고리즘과는 달리 700여 개의 빔들이 지연 시간 없이 요청된 시간에 처리되고 있으며, 전체적으로 처리된 분포를 봐도 제때 처리된 빔들의 개수가 가장 많다. 또 다른 한 가지 특징은 요청된 시간보다 미리 처리되는 빔들도 존재한다는 것이다. 레이더의 dwell time이 길고, 짧은 시간 안에 많은 수의 추적 임무들이 요청되는 과부하 상태에서는 빔들의 지연 시간이 레이더의 전체 성능을 크게 저하할 수 있는데, 요청된 시간보다 일

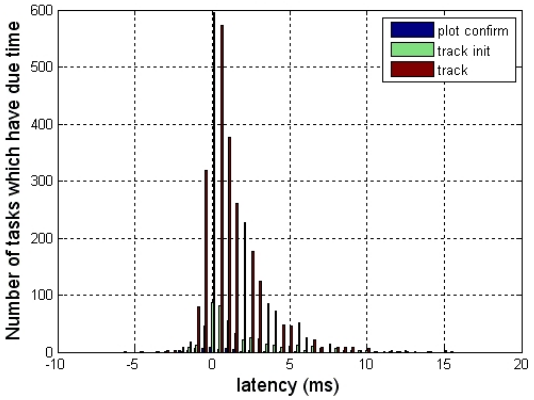


(a) Butler scheduler에서 빔별 시간 지연도
(a) Latency according to beam type in Butler scheduler

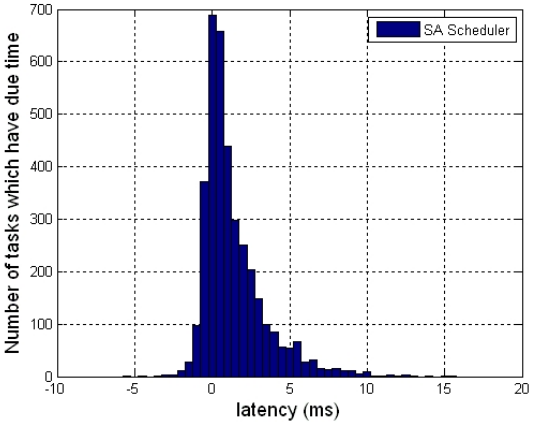


(b) Butler scheduler에서 전체 시간 지연도
(b) Total latency in Butler scheduler

그림 1. Butler scheduler를 적용한 빔 스케줄링 결과
Fig. 1. Scheduling result by Butler scheduler.



(a) SA scheduler에서 빔별 시간 지연도
(a) Latency according to beam type in SAr scheduler



(b) SA scheduler에서 전체 시간 지연도
(b) Total latency in SA scheduler

그림 2. SA scheduler를 적용한 빔 스케줄링 결과
Fig. 2. Scheduling result by SA scheduler.

찍 처리되는 빔을 허락함으로써 비슷한 시간에 요청되는 빔들 간의 충돌을 해소할 수 있다는 장점이 있다.

그림 3은 동일한 실험 시나리오에서 Butler 알고리즘을 적용했을 때와 SA를 적용한 빔 스케줄링 결과에 대한 빔 처리 지연도를 하나의 그림에서 비교한 것으로, 앞서 설명한 것처럼 SA를 적용한 것이 전체적으로 많은 빔들을 제 때에 처리하는 것을 알 수 있다.

4-2-2 스케줄링한 빔의 개수 비교

스케줄링 알고리즘의 성능 분석을 위해 처리된 빔의 개수도 중요한 성능 분석 요소 중의 하나이다. 표 4는 알고리즘별 시뮬레이션 시간 및 출현하는 표적의 수를 달리하여 반복 실험을 통해 처리된 빔의 평균 개수를 구한 것이다. 먼저 30초의 시뮬레이션 시간 동안 표적이 50개 출현하였을 경우(그림 2와 그림 3의 결과에 적용된 동일한 시나리오) SA를 적용한 스케줄링 결과가 평균 198개의 빔을 주어진 시간 동안 더 많이 처리했음을 알 수 있다. 표적의 개수를 동일하게 유지하고 시뮬레이션 시간을 50초로 길게 한 경우에도 SA가 처리한 빔의 개수가 348개 더 많다. 또한, 시뮬레이션 시간을 동일하게 유지하고 출현하는 표적의 수를 증가한 경우에도 유사하게 평균 294개의 빔을 더 처리하였음을 확인하였다. 이는 SA를 적용한 빔 스케줄러가 동일한 전장 환경

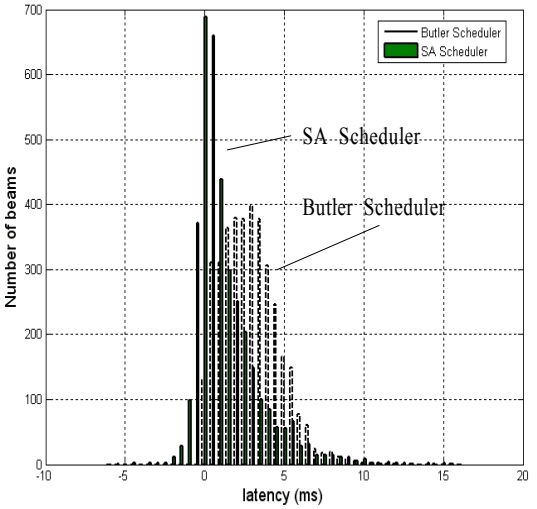


그림 3. Butler scheduler와 SA scheduler의 빔 처리 지연도 결과 비교

Fig. 3. Comparison of beam processing latency between Butler scheduler and SA scheduler.

표 4. 알고리즘 별 스케줄링한 빔의 평균 개수

Table 4. The average number of processed beams according to each algorithm.

	SA scheduler(1)	Butler scheduler(2)	(1)－(2)
N:50개 T:30s	12,006	11,808	+198
N:50개 T:50s	19,903	19,555	+348
N:80개 T:30s	12,467	12,173	+294

N: 시뮬레이션한 표적의 수, T: 시뮬레이션 수행 시간

에서 시간이 지날수록 처리할 수 있는 빔의 개수가 더 많아지는 동시에, 동일한 시간에도 더 많은 표적의 탐지 및 추적이 가능함을 의미한다. 주어진 시간에 더 많은 빔을 처리할 수 있다는 것은, 레이더가 주어진 시간 안에 더 많은 임무를 처리할 수 있다는 뜻이며, 이는 레이더의 자원을 효율적으로 활용하기 위한 레이더 자원 관리 기술의 궁극적인 목표이기도 하다.

4-2-3 Frame Time 비교

그림 4는 Butler 알고리즘을 적용했을 때와 SA를 이용한 스케줄링 알고리즘을 적용했을 때의 frame time의 변화를 보여준다. 50개의 표적이 20초 내에 서 출현하며, 30초 동안 시뮬레이션한 결과로 두 알고리즘이 공통적으로 frame time이 계속 증가하다가 20초를 기준으로 frame time이 일정하게 유지된다. 두 알고리즘에서 frame time이 조금 차이가 나는데, Region 2에서는 SA 알고리즘 적용결과가 미비하게 약간 더 높게 소요되었지만, 그 차이기 크지 않고 거의 유사하다. 반면, Region 1에서는 모든 표적이 출현한 이후 시간을 보면, SA를 적용했을 때의 frame time이 Butler를 적용했을 때의 frame time보다 0.5초 이상으로 짧게 소요되었다. Frame time이 짧다는 의미는 동일 공간을 전체적으로 한 번 탐지하는데 소요되는 시간이 짧다는 것이고, 주어진 시간 내에 레이더에게 탐지를 위해 가용한 시간을 더 많이 허락할 수 있다는 의미이다. 이는 또한 앞서 언급한 SA 적용시 스케줄링한 빔의 평균 개수에서의 우월성과도 밀접한 관련이 있는데, 주어진 시간 내에 많은 빔을 처리한다는 것은 동일한 크기의 영역을 더 빨리 처리할 수 있다는 의미이기 때문이다.

4-2-4 실시간성을 보장하기 위한 스케줄링 수행 시간 비교

위의 성능 분석 결과와 같이, 빔 처리 시간 지연도, 빔 처리 개수, frame time 측면에서 SA를 적용한 빔 스케줄링 결과가 기존의 방법론에 비해 성능이 우수함을 확인하였다. 하지만 이런 성능 개선과 더불어 실제 레이더 시스템에 적용하기 위해서는 반드시 실시간성이 보장되어야 한다. 표 5는 현재 시뮬레이션 시나리오에서 스케줄링 시간과 관련된 정보를 보여준다. Scheduling_Time_Interval을 10 ms로 했을 때 SA 스케줄러가 한 번에 순서를 최적화하는 빔의 개수가 평균 7.24개였으며, 이때 스케줄링에 소요되는 시간이 2.9 ms 소요되었다. 즉, 10 ms 주기로 2.9 ms를 소요하여 평균 7.24개의 빔을 스케줄링하게 되는 것이다. 이는 다음 스케줄링까지의 여유시간을 고려할 때, 실시간성을 보장하며 실제 레이더 시스템에 적용 가능함을 의미한다.

하지만, 추가적으로 고려해야 할 사항은 이러한 스케줄링 소요 시간은 SA 알고리즘의 종료 조건에에 의해 결정되며, 이러한 종료 조건은 알고리즘이 찾는 해의 최적화 정도와 trade-off 관계에 있다는 것

표 5. SA 적용시 스케줄링 소요 시간
Table 5. Scheduling time of SA scheduler.

Scheduling_Time_Interval	평균 빔의 개수	소요 시간
10 ms	7.24개의 빔	2.9 ms

수행 시간 측정 환경: Intel(R) Core(TM)2 Quad CPU Q8300 @2.50 GHz, 3.50 GB RAM, 구현 언어: Matlab R2009b

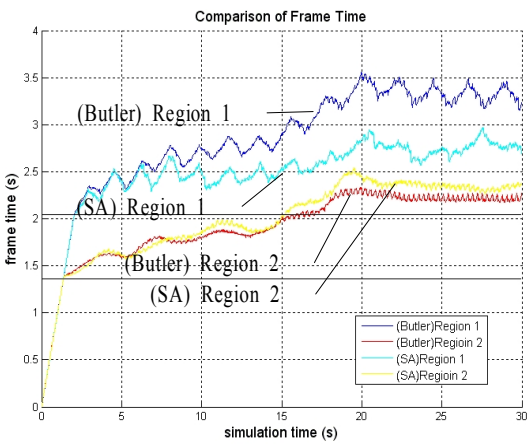


그림 4. Butler scheduler와 SA scheduler의 frame time 비교
Fig. 4. Comparison of frame time between Butler scheduler and SA scheduler.

이다. 예를 들어 Scheduling_Time_Interval이 길어서 한 번에 처리할 수 있는 빔의 개수가 많아질 경우, 수렴 속도가 늦어져서 동일한 종료 조건을 가질 경우 해가 최적화 되지 않을 수 있고, 종료 조건을 바꿀 경우 스케줄링 소요 시간이 길어질 수 있다. 하지만 실제로 Scheduling_Time_Interval이 길어지면 오랜 시간 동안의 Scheduling_Time_Interval 시간 내에서 발생하는 다른 이벤트에 대한 즉각적인 반응이 어려워지는 문제가 있을 수 있다. 따라서 필요 이상으로 Scheduling_Time_Interval을 길게 설정하는 것은 수렴시간 문제를 차치하더라도 바람직하지 않다.

본 논문의 시험 결과를 통해 알 수 있듯이 SA를 이용한 스케줄링을 실제 레이더 시스템에 적용할 경우, 실시간성을 보장하면서 최적화된 성능을 내기 위해서는, 해당 시스템에서 빔의 평균 dwell time을 고려하여 한 번에 최적화할 빔의 개수를 평균 7개 정도로 하는 Scheduling_Time_Interval을 잡는 것이 바람직할 것이다. 참고로 Butler 알고리즘을 적용했을 경우 매번 하나씩의 빔을 스케줄링하게 되며, 한 번의 스케줄링에 소요되는 시간이 0.1 ms로 측정되었다.

Ⅴ. 맺음말

본 논문에서는 레이더 자원 관리의 핵심 요소라고 할 수 있는 레이더 빔 스케줄링을 위해 SA를 적용 추계적 레이더 빔 스케줄러 구현에 관한 내용을 제안하였고, 이를 기존 방법론과 비교해 보았다. 특히, 빔의 처리 시간 지연도와, 동일한 시간에서 스케줄링한 빔의 개수를 비교해 보았을 때 그 성능이 우월함을 입증하였다. 이는 기존의 방법에 비해 동일한 전장 환경에서 시간이 지날수록 처리할 수 있는 빔의 개수가 더 많아지는 동시에, 동일한 시간에도 더 많은 표적을 탐지, 추적할 수 있음을 의미한다. 주어진 시간에 더 많은 빔을 처리할 수 있다는 것은, 레이더가 주어진 시간 안에 더 많은 임무를 처리할 수 있다는 뜻이며, 궁극적으로는 효율적인 레이더 자원 활용에 크게 도움을 줄 수 있을 것이다. 또한 성능 향상과 더불어 스케줄링 수행 시간 평가를 통해 그 실시간성을 확인하였다. 향후 연구로는, 추적 필터를 연동하여 표적의 추적 정확도의 측면에서 스케줄링 알고리즘을 보완하고, 다양한 성능 지수를 통해 보다 정교한 성능 평가를 수행할 필요가 있다.

참 고 문 헌

[1] Zhen Ding, "A survey of radar resource management algorithms", *IEEE Canadian Conference on Electrical and Computer Engineering*, *CCECE*, pp. 1559-1564, 2008.

[1] Zhen Ding, "A survey of radar resource management algorithms", *IEEE Canadian Conference on Electrical and Computer Engineering*, *CCECE*, pp. 1559-1564, 2008.

[2] J. M. Butler, Multi-function radar tracking and control, University College London. Ph.D. Thesis, 1998.

[3] A. J. Orman, C. N. Potts, A. K. Shahani, and A. R. Moore, "Scheduling for a multifunction array radar

system", *European Journal of Oparational Research*, no. 90, pp. 13-25, 1996.

[4] S. L. C Miranda, C. J. Baker, K. Woodbridge, and H. D. Griffiths, "Phased array radar resource management: a comparison of scheduling algorithms", in *Proc. IEEE Radar Conf.*, pp. 79-84. 2004.

[5] W. Komorniczak, J. Pietrasinski, "Selected problems of MFR resources management", *the 3rd International Conference on Information Fusion*, vol. 2, pp. 3-8, Jul. 2000.

[6] S. L. C. Miranda, K. Baker, K. Woddbridge, and H. D. Griffiths, "Fuzzy logic approach for prioritisation of radar tasks and sectors of surveillance in multifunction radar", *IET Radar Sonar Navigation*, vol. 1, no. 2, pp. 131-141, 2007.

[7] V. Krishnamurthy, R. J. Evans, "Hidden Markov model multiarm bandits: A methodology for beam scheduling in multitarget tracking", *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 2893-2908, 2001.

[8] S. Ghosh, R. Rajkumar, J. Hansen, and J. Lehoczky, "Integrated QoS aware resource management and scheduling with multiresource constraints", *Real Time System*, vol. 33, pp. 7-46, 2006.

[9] S. Gopalakrishnan, C. S. Shih, P. Ganti, M. Caccamo, and L. Sha, "Radar dwell scheduling with temporal distance and energy constraints", *International Radar Conference*, 2004.

[10] K. Harada, T. Ushio, and Y. Nakamoto, "Adaptive resource allocation control for fair QoS management", *IEEE Transactions on Computers*, vol. 56, no. 3, pp. 344-357, 2007.

[11] 한순미, 백승우, 조선영, 조겸래, 이대우, 김해동, "유전 알고리즘을 이용한 위성 임무 스케줄링 최적화", 한국항공우주학회지, 36(12), pp. 1130-1239, 2008.

[12] M. L. Pinedo, *Scheduling Theory*, *Algorithms*, *and Systems*, Third Edition, Springer, ISBN: 978-0-387-78934-7, 2008.

[13] Stefka Fidanova, "Simulated annealing for grid scheduling problem", *IEEE John Vincent Atanasoff International Symposium on Modern Computing(JVA'06)*, pp. 41-45, 2006.

[14] V. Cerny, "A thermodynamical approach to the traveling salesman problem: An efficient simulated annealing algorithm", *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41-51, 1985.

[15] K. Dowsland, "Variants of simulated annealing for practical problem solving", V. Rayward-Smith editor, *Applications of Modern Heuristic Methods*, Henley-on-Thames: Alfred Walter Ltd., 1995.

[16] I. H. Osman, C. N. Potts, "Simulated annealing for permutation flow-shop scheduling", *Omega 17*, pp. 551-557, 1989.

[17] 강승택, "Ka-대역 군지연-등화 여파기용 SA 기법과 CG 기법의 하이브리드 설계 기법", 한국전자파학회논문지, 15(8), pp. 775-780, 2004년.

노 지 은

2002년 2월: 포항공과대학교 컴퓨터공학과 (공학석사)
2006년 2월: 포항공과대학교 컴퓨터공학과 (공학박사)
2006년 2월～현재: 국방과학연구소 선임연구원
[주 관심분야] 레이더 신호처리, 통제 알고리즘 등

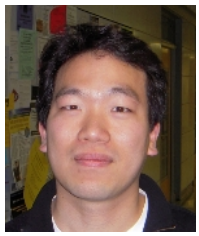
안 창 수

2005년 2월: 고려대학교 전파통신공학과 (공학석사)
2005년 3월～현재: 국방과학연구소 선임연구원
[주 관심분야] 반도체 송수신 모듈 설계, 항공기 레이더 운용모드 설계, 능동 위상 배열 레이더 시스템 설계

김 선 주

1988년 2월: 아주대학교 전자공학과 (공학석사)
1988년 3월～현재: 국방과학연구소 책임연구원
[주 관심분야] 반도체 송수신 모듈 설계, 능동 위상 배열 레이더 시스템 설계, 항공기 레이더

최 한 림



2000년 2월: 한국과학기술원 항공우주공학과 (공학사)
2002년 2월: 한국과학기술원 항공우주공학과 (공학석사)
2009년 2월: MIT Aero/Astro (Ph.D.)
2002년 3월～2004년 7월: PSIA㈜
2008년 10월～2010년 2월: MIT Aero/Astro (Postdoctoral Associate)
2010년 3월～현재: 한국과학기술원 항공우주공학과 조교수
[주 관심분야] 레이더 자원관리, 추정 이론, 협업 제어, 로보틱스, 최적화

장 대 성

2008년 2월: 한국과학기술원 항공우주공학과 (공학사)
2008년 3월～현재: 한국과학기술원 항공우주공학과 석박사 통합과정
[주 관심분야] 레이더 자원관리, 복수 에이전트 임무 할당/스케줄링, 인공지능