

TB와 냅색 기반의 향상된 다기능 레이더 스케줄링 기법

TB and Knapsack Based Improved Scheduling Techniques for Multi-Function Radar

황민영 · 양우용* · 신상진* · 전주환

Min-Young Hwang · Woo-Young Yang* · Sang-Jin Shin* · Joohwan Chun

요 약

현대의 레이더는 위상 배열 안테나로 전자기파 빔을 생성하여 여러 가지 작업을 수행한다. 레이더가 수행하는 작업으로는 목표물의 탐지와 추적, 미사일 유도 등이 있다. 다기능 레이더가 개발되기 이전에는 레이더는 한 가지 작업만을 수행할 수 있기에, 각 작업을 수행하는 독립된 레이더가 필요했다. 하지만 다기능 레이더는 다양한 작업을 총괄하여 수행할 수 있는 기능을 탑재하고 있기 때문에, 하나의 레이더로 필요한 여러 작업을 수행할 수 있다. 레이더의 한정된 시간 자원을 효율적으로 이용하기 위해 예약된 작업을 적당한 순서로 나열하는 기법을 스케줄링이라고 한다. 본 연구에서는 두 가지의 스케줄링 기법들을 파악하여 효율적인 스케줄링 기법을 제시하여 보았다.

Abstract

Modern radars such as the phase array radar can handle various tasks by generating a beam from a phased array antenna. Radar can be used for miscellaneous applications such as surveillance, tracking, missile guidance etc. Previous radar systems could handle only one task at a time. As such, multiple radars were required to perform simultaneous tasks. Multi-function radars can perform many tasks using only one radar system. However, the radar's resources are limited in this instance. To efficiently utilize time, it is necessary to properly schedule tasks in the radar's timeline. In this report, we investigate the efficiency of different scheduling tasks.

Key words: Phased Array Radar, Multi-Function Radar, Radar Resource Management, Scheduling

I. 개 요

레이더는 강한 전자기파를 이용하여 물체를 탐지한다. 일반적인 레이더는 안테나를 360도 회전하며 전자기파를 방사하고, 물체에 반사된 전자기파를 통해 목표물의 위치를 파악한다. 하지만 위상 배열 안테나 레이더(phased array

radar: PAR)는 기계적으로 안테나를 회전시키며, 전자기파를 방사할 필요 없이 고정된 안테나에서 방사하는 전자기파의 위상을 바꾸는 방법으로 전자기파 빔을 만들어 원하는 지점을 살필 수 있다.

레이더는 목표물(target)의 탐지(surveillance)와 추적(tracking), 미사일 유도(missile guidance) 등의 다양한 작업(task)

「이 연구는 한화시스템즈의 지원으로 연구되었음.」

한국과학기술원 전기 및 전자공학부(Department of Electrical Engineering, Korean Advanced Institute of Science and Technology)

*한화시스템즈(Hanhwa Systems)

· Manuscript received August 16, 2018 ; Revised September 7 2018 ; Accepted December 18, 2018. (ID No. 20180816-083)

· Corresponding Author: Min-Young Hwang (e-mail: robco@kaist.ac.kr)

을 수행하는 데 이용된다. 기존의 레이더는 하나의 시스템에서 한 종류의 작업만을 수행할 수 있었기에, 각각의 작업에 대해 독립된 레이더 시스템이 필요했다. 따라서 군용 함정과 같이 레이더를 통해 여러 가지 작업을 수행하여야 하는 경우에는 다수의 레이더를 탑재하고 있어야만 하였다. 하지만 다기능 레이더(multi-function radar: MFR)는 여러 작업을 하나의 레이더 시스템으로 총괄하여 처리할 수 있어 여러 대의 레이더를 설치할 필요가 없어진다.

다기능 레이더는 아주 유용하게 활용될 수 있지만, 그 자원이 제한되어 있다. 따라서 제한된 레이더의 자원을 효율적으로 이용하는 기술이 필요하다. 그중에서도 스케줄링(scheduling)은 레이더의 시간 자원을 효율적으로 활용하기 위해 레이더가 수행하여야 할 여러 가지 작업을 적당한 순서로 나열하는 기술이다.

스케줄링에는 다양한 방법이 존재한다. 가장 기본적인 방법은 작업에 대해 우선순위를 매겨 나열하는 방법이다. 예를 들어 탐지, 추적 그리고 미사일 유도 작업이 수행되어야 한다면, 가장 우선순위가 높은 미사일 유도 작업이 가장 먼저 스케줄링 되고, 탐지 그리고 추적 순으로 스케줄링이 될 것이다. 이 방법은 간단하고 명료하지만, 다양한 환경적인 변화를 반영하지 못한다. 따라서 비효율적으로 레이더 시간 자원이 이용될 가능성이 있다. 따라서 레이더가 수행할 작업의 속성을 파악하고, 적절한 스케줄링 기법을 사용하는 것이 중요하다.

본 연구에서는 먼저 다기능 레이더 모델에 대해 간략히 설명하고, 그러한 시스템에서 어떤 스케줄링 기법이 가장 효율적으로 레이더 시간 자원을 이용할 수 있는지를 알아보려 한다. 제시된 모델이 실제 환경을 완벽하게 고려하지는 못하지만, 스케줄링 기법의 특성을 파악하고 비교할 수는 있을 것이다. 스케줄링 시뮬레이션은 ‘MATLAB’에서 구현하였다.

II. 다기능 레이더 모델

스케줄링에 앞서 먼저 다기능 레이더 시스템 모델에 대하여 알아보겠다.

2-1 레이더 작업(Radar Task)

다기능 레이더는 기본적으로 목표물의 위치를 파악하

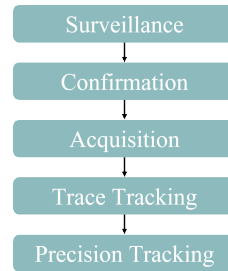


그림 1. 레이더 작업 수행 과정

Fig. 1. Radar task process.

는 탐지 작업(surveillance task), 목표물이 파악되면 계속해서 그것의 위치를 추적하는 추적 작업(tracking task), 그리고 자신이 쏘아 올린 미사일의 방향을 목표물로 안내하는 미사일 유도 작업(missile guidance task)을 수행할 수 있다.

레이더가 목표물을 파악하여 작업을 수행하는 과정은 그림 1로 표현할 수 있다. 먼저 레이더는 아무런 목표물이 설정되지 않았을 때는 탐지 작업을 수행한다. 위상 배열 안테나가 전자기파 빔을 만들어 원하는 지점을 탐지해 나간다. 그리고 목표물로 파악되는 물체가 발견되면 확인 작업(confirmation task)과 습득 작업(acquisition task)을 통해 탐지한 물체가 추적해야 할 목표물인지를 판단한다. 물체가 목표물로 설정되면 자취 추적 작업(trace tracking task)을 시작한다. 추적 작업이 시작되면 레이더는 일정한 시간마다 추적하는 목표물에 위치에 전파를 보내 위치를 추적한다. 그리고 만약에 추적하던 목표물이 아주 중요하거나, 추적이 응급한 상황이라면 정밀 추적 작업(precision tracking task)을 시작한다. 정밀 추적 작업은 미사일 유도 작업과 같은 작업으로 다른 모든 작업보다 우선으로 수행되어야 한다.

이러한 모든 과정은 하나의 레이더 시스템에서 이루어지기 때문에 문제가 생길 수 있다. 예를 들어 추적에 너무 집중하면 물체를 탐지하지 못하게 되고, 탐지에 레이더를 길게 사용한다면 추적하던 목표물을 잃어버릴 수도 있다. 따라서 레이더 작업 스케줄링을 통해 효율적으로 작업의 순서를 정렬하는 것이 중요하다.

2-2 작업 인터리빙(Task Interleaving)

레이더 작업 스케줄링에는 작업 인터리빙이라는 중요

한 개념이 있다. 레이다는 전파를 방사하고 반사되어 돌아오는 전파를 분석하여 물체의 위치를 찾는다. 따라서 추적 작업과 같이 지속해서 물체의 위치를 파악하기 위해 전파를 보내고 받는 작업은 그림 2와 같이 송신(transmitting) 그리고 수신(receiving)의 두 부분으로 나눌 수 있다. 송신과 수신 사이에 레이다가 대기하는 시간을 대기 시간(idle time)이라고 한다. 보통 송신 시간과 수신 시간은 목표물에 전파하는 전자기파 빔의 파워 크기와 관련이 있다. 크기가 작거나, 스텔스 기능이 있는 목표물일수록 탐지하기 위해서는 더 큰 파워가 필요하고, 이로 인해 송수신시간이 길어지게 된다. 대기시간은 목표물과 레이다 간의 거리에 비례한다.

우리는 레이다가 아무런 작업을 수행하지 않고 기다려야 하는 시간인 대기시간을 줄이고자 작업 인터리빙을 한다. 대기시간에 다른 작업을 끼워 넣어 레이다가 대기하는 시간을 줄여 좀 더 레이다의 타임라인(timeline)을 뾰뾰하게 구성하고자 하는 것이다.

그림 3에서 “Surv”는 탐지 작업을 의미하고, “Task n”은 n번째 목표물에 대한 추적 작업을 의미한다. 인터리빙이 없는 경우에는 대기시간에 레이다가 아무런 작업을 진행하지 않고 있는 것을 볼 수 있다. 하지만 작업 인터리빙이 있는 경우에는 어떠한 작업의 송신 부분과 수신 부분 사이에 다른 작업이 끼여 들어가 있어, 같은 시간 동안에 더 많은 작업을 진행 중인 것을 볼 수 있다.

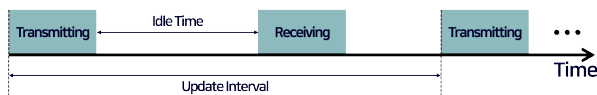


그림 2. 작업 인터리빙 예시
Fig. 2. Task interleaving.

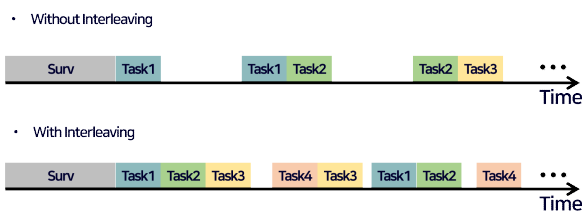


그림 3. 작업 인터리빙 예시
Fig. 3. Task interleaving example.

2-3 갱신 주기(Update-rate)

추적 작업과 같이 지속해서 목표물의 위치를 파악하여야 하는 작업의 경우에는 일정한 주기를 갖고 작업을 반복적으로 수행한다. 그것은 갱신 주기라고 하며, 갱신 시간(update time)마다 작업이 반복되게 된다. 일반적으로 목표물의 이동 속도가 빠르다면 갱신 시간이 짧아져 더 자주 작업이 실행되게 된다. 그림 2에서 update interval이 갱신 시간과 같은 의미이다.

III. 레이다 스케줄링 기법

레이다 스케줄링 기법에는 여러 가지가 있다. 이번 연구에서는 가장 기초적이면서도 단순한 방법인 타임 밸런스 스케줄링(time-balance scheduling) 기법과 탐색 스케줄링(knapsack scheduling)에 대해 주로 알아보았고, 그 문제점이 보완하려는 방법을 생각해 보았다.

3-1 타임 밸런스 스케줄러(Time-Balance Scheduler)

타임 밸런스(time-balance: TB)는 어떠한 작업이 얼마만큼의 시간 동안 스케줄링되지 못하였는지를 의미한다. TB 값은 기본적으로 시간이 지날수록 일정하게 커진다. TB 값이 음수인 경우에는 아직 작업이 스케줄링될 준비가 되지 않은 것이다. TB 값이 0인 순간에는 그 작업이 즉시 스케줄링되어야 한다. 하지만 다른 작업이 우선으로 스케줄링되어야 하여 바로 스케줄링 되지 못한다면, TB 값은 0에서부터 계속 증가해 양수가 될 것이고, 그만큼의 시간을 스케줄링되지 못하고 밀리게 된 것이다. 결국 여러 작업 중에 가장 큰 TB 값을 가진 것부터 스케줄링된다. 작업이 한번 스케줄링되면 TB 값은 그 작업이 가진

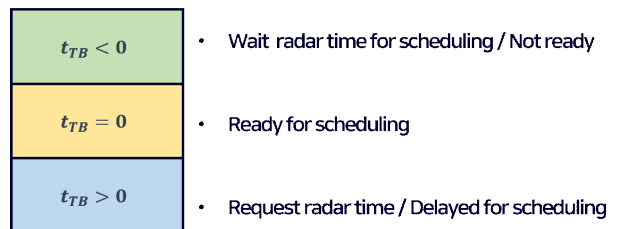


그림 4. 타임 밸런스
Fig. 4. Time-balance.

갱신 시간(update time)만큼 줄어들게 된다.

탐지 작업은 항상 그 우선도가 다른 작업보다 낮다. 따라서 TB 값이 따로 주어지지 않고, 다른 작업의 TB 값이 모두 음수일 때 탐지 작업이 실행된다.

3-1-1 타임 밸런스 스케줄러 시뮬레이션 1

TB 스케줄러의 시뮬레이션은 두 가지 상황에 대해 진행하기로 했다. 첫 번째는 각 작업이 한 번씩만 반복되는 상황이고, 두 번째는 각 작업이 주어진 갱신 시간(update time)을 가져 일정한 시간마다 반복해서 수행되는 상황이다. 먼저 첫 번째 상황에 대한 시뮬레이션 1을 진행하였다.

시뮬레이션 1을 위해 그림 5와 같이 다섯 개의 추적 작업을 가정했다. 각각의 작업들은 주어진 수행 시간(processing time) 동안 작업을 수행하고, 대기시간이 지난 후에 한 번 더 작업을 수행하게 된다. 모든 작업들의 TB 값은 0부터 시작했고, TB 값이 같은 경우에는 작업의 번호가 작은 순서대로 우선 순위를 주었다. 그래프에서 가로축의 단위는 단위시간으로 가정했고, 레이다의 총 타임라인의 길이는 30단위시간으로 설정하였다.

3-1-2 타임 밸런스 스케줄러 시뮬레이션 1 결과

그림 6은 주어진 작업들의 TB 스케줄링 결과이다. 점유도(occupancy)는 전체 타임 슬롯(time slot)이 비어 있지

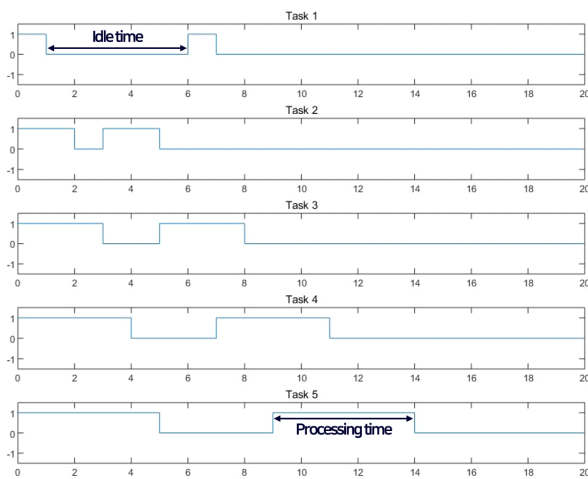


그림 5. 타임 밸런스 스케줄러 시뮬레이션 1 작업
Fig. 5. TB scheduler simulation 1 tasks.

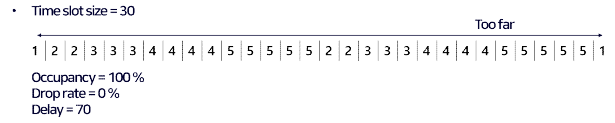


그림 6. 타임 밸런스 스케줄러 시뮬레이션 1 결과
Fig. 6. Results of TB scheduler simulation 1.

않고 얼마만큼 이용되고 있는지를 나타내는 값이다. 결과에서 타임 슬롯은 빈 부분 없이 완전히 활용되고 있는 것을 볼 수 있다. 탈락률(drop rate)은 작업 중에 시간의 부족으로 스케줄링되지 못하고 탈락한 것이 있는지를 나타내는 값이다. 결과 값에서는 작업들 중에 탈락한 것은 없다. 문제는 지연(delay) 값이다. 이 값은 작업이 이상적인 대기시간에서부터 얼마나 이후에 다시 스케줄링되었는지를 나타내는 값이다. 결과물을 보면 Task 1은 가장 먼저 스케줄링되어있다. 그리고 이상적으로는 Task 1은 5만큼의 단위시간 이후에 다시 스케줄링되어야 한다. 하지만 TB 스케줄러의 특성상 TB 값이 더 큰 작업이 먼저 스케줄링되어야 하기 때문에, 두 번째 Task 1은 한참 뒤에 스케줄링되어있다. 이렇게 이상적으로 스케줄링되어야 하는 시간보다 떨어진 정도를 지연 값으로 계산하였다. 다른 Task들도 조금씩 지연된 채 다시 스케줄링 되어있고, 그 밀린 정도를 전부 합한 값이 무려 70이나 된다. 즉, 기본적인 TB 스케줄러는 시간 효율적으로 스케줄링할 수 있지만, 안정적으로 작업을 수행하기 힘들어진다.

3-1-3 타임 밸런스 스케줄러 시뮬레이션 2

시뮬레이션 2를 위해 그림 7에서와같이 세 개의 추적

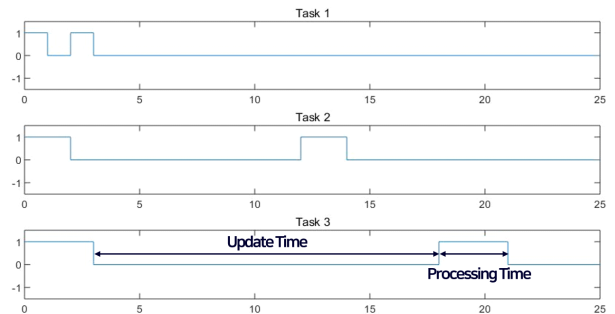


그림 7. 타임 밸런스 스케줄러 시뮬레이션 2 작업
Fig. 7. TB scheduler simulation 2 tasks.

작업을 가정했다. 각각의 작업들은 주어진 수행시간 동안 작업을 수행하고, 갱신 시간마다 작업을 반복해서 수행하게 된다. 레이다의 총 타임라인의 길이는 50단위 시간으로 설정하였다.

3-1-4 타임 밸런스 스케줄러 시뮬레이션 2 결과

그림 8은 시뮬레이션 2의 TB 스케줄링 결과이다. 이번 시뮬레이션에서 작업은 반복해서 수행되기 때문에 탈락률은 고려하지 않았다. 점유도는 최대로 나타나고 있지만, 지연 값은 여전히 크게 나타나고 있다는 것을 확인할 수 있다. 이것은 작업이 적절한 시간에 수행되고 있지 못함을 의미한다.

3-2 냅색 스케줄러(Knapsack Scheduler)

냅색 스케줄러는 흔히 냅색 문제(Knapsack problem)로 알려진 문제를 해결하는 알고리즘을 이용한 스케줄러이다.

냅색 문제를 간략하게 설명해 보면 다음과 같다. 일정한 용량(capacity)을 가지는 배낭이 있고, 각각 가치(value)와 무게(weight)가 다른 물건들이 있다. 주어진 배낭의 무게 한도 내에서 가치가 최대가 되도록 물건을 배낭에 담고자 한다. 이 문제를 해결하는 알고리즘이 냅색 알고리즘이다. 그림 9에서 배낭의 용량은 10 kg이다. 이때 가치를 최대로 하도록 물건을 담는 방법은, “가치/무게”가 가장 큰 순서대로 물건을 담는 것이다. 따라서 배낭 안의 물건의 가치가 최대가 되도록 하려면 핸드폰, 책 그리고 로봇 순서대로 담아야 할 것이다. 실제 프로그래밍에서는 좀 더 발견적인 동적 프로그래밍을 통해 결과를 구한다.

냅색 스케줄러는 두 단계로 나뉜다. 첫 번째 단계에서는 제한된 시간 동안에 작업들의 우선도 합이 최대가 되도록 수행할 작업을 추려낸다. 이 때 냅색 알고리즘을 이용한다. 여기서 냅색의 용량은 타임라인의 길이와 같다.

• Time slot size = 50
 1 | 2 | 2 | 3 | 3 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 3 | 3
 3 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 3 | 1 | 1
 Occupancy = 100%
 Delay = 12

그림 8. 타임 밸런스 스케줄러 시뮬레이션 2 결과
 Fig. 8. Results of TB scheduler simulation 2.

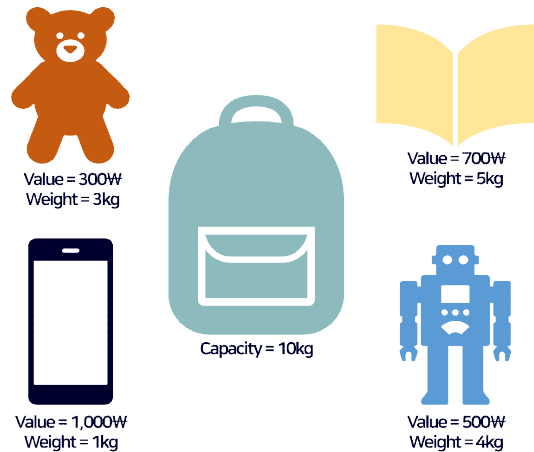


그림 9. 냅색 문제
 Fig. 9. Knapsack problem.

그리고 물건의 가치는 작업의 우선순위와 같고, 물건의 용량은 작업의 실행 시간과 같다. 제한된 시간 내에 실행될 작업을 추려내면, 두 번째 단계에서 작업의 우선순위에 따라 레이다 타임라인에 나열한다. 시뮬레이션에서는 첫 번째 단계까지만 진행하였다.

3-2-1 냅색 스케줄러 시뮬레이션

냅색 스케줄러는 먼저 작업의 개수와 우선순위 그리고 실행 시간이 결정되어 있어야 한다. 먼저 그림 10과 같이

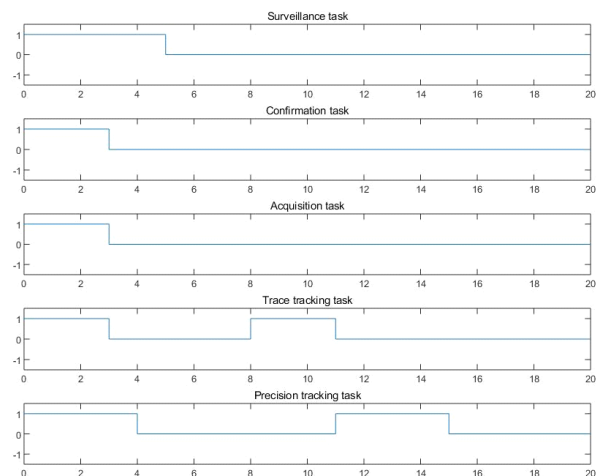


그림 10. 냅색 스케줄러 시뮬레이션 작업
 Fig. 10. Knapsack scheduler simulation tasks.

다섯 종류의 작업을 설정하였다. 우선도(priority)는 탐색 문제에서의 물건의 가치와 같은 값으로 우선도 값이 클수록 작업의 우선순위가 높은 것이다. 우선도 값으로 탐지 작업은 1, 확인 작업은 2, 습득 작업은 3, 자취 추적 작업은 4 그리고 정밀 추적 작업은 5의 값을 주었다.

다음으로는 작업의 개수를 설정해 주었다. 작업의 개수를 초기에 고정적으로 설정하지 않고 확률적으로 생성해 내기로 하였다. 따라서 총 n 개의 작업이 있을 때, 각 작업의 종류별로 몇 개가 존재할지 결정하기 위해서 마르코프 체인을 이용하였다. 그림 11과 같이 마르코프 체인을 작성하였다. 각각의 작업 단계들은 가지들을 가지고 있다. 각 가지는 해당 단계에서 다음 단계로 넘어갈 확률을 갖고 있고, 그 확률값은 그림 11에서와 같이 설정해 주었다. 예를 들어 습득 작업 단계는 자기 자신으로 돌아올 확률(0.2), 탐지 작업으로 돌아갈 확률(0.3) 그리고 자취 추적 작업으로 넘어갈 확률(0.5)의 세 가지 확률들이 있다. 화살표가 이어지지 않은 작업으로 넘어갈 확률은 없다. 모든 가지의 확률의 합은 1이 된다. 이렇게 다음 작업으로 넘어갈 확률들을 행렬로 표현한 다음, 초기 작업의 존재 확률값에 반복해서 곱해 주면, 각각 작업이 존재할 확률로 수렴하게 된다. 예를 들어 총 10개의 작업이 생성한다고 하였을 때, 마르코프 체인으로 탐지 작업의 존재 확률이 0.3으로 수렴했다고 하면, 10개 중에 3개의 작업은 탐지 작업으로 생성된다. 따라서 하나의 레이더 타임라인 동안에 탐지 작업은 3번 반복되어 수행되어야 하는 것이다.

$$M = \begin{bmatrix} 0.5 & 0.4 & 0.3 & 0.3 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0.2 & 0 & 0 \\ 0 & 0 & 0.5 & 0.3 & 0 \\ 0 & 0 & 0 & 0.4 & 0.5 \end{bmatrix}, p = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}, P = \begin{bmatrix} 0.4242 \\ 0.2121 \\ 0.1591 \\ 0.1136 \\ 0.0909 \end{bmatrix}$$

시뮬레이션에서 활용한 마르코프 체인 행렬은 위에서 나타낸 바와 같다. 초기 확률 p 에 마르코프 체인 행렬 M 을 반복해서 곱해 주면 결국 작업이 존재할 확률 P 의 값에 수렴하게 된다.

시뮬레이션에서 총 작업의 수는 20개로 설정하였다. 그리고 레이더 타임라인의 길이는 40단위시간 그리고 60 단위시간으로 설정하였다. 마르코프 체인으로 만들어낸 확률로 작업의 종류를 결정하고, 생성된 작업을 탐색 스

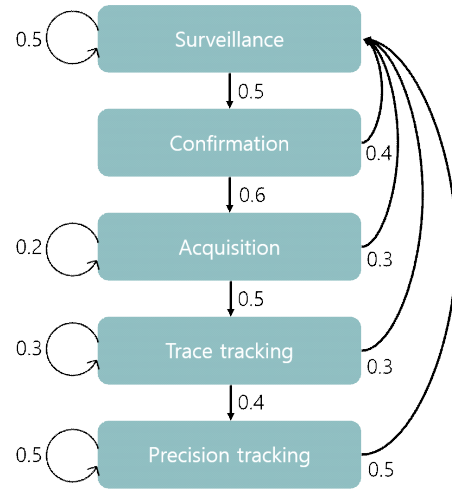


그림 11. 레이더 작업 마르코프 체인
Fig. 11. Markov chain for radar task.

케줄러로 제한된 시간 내에 최대의 우선도 값이 나오도록 하였다.

3-2-2 탐색 스케줄러 결과

탐색 스케줄러로 주어진 환경에서 시뮬레이션한 결과, 그림 12와 같은 결과를 얻을 수 있었다. 탐색 스케줄러는 처음에 주어진 시간 내에 실행할 우선도가 가장 높은 작업을 빠르게 분류해준다는 장점이 있다. 하지만 여기서 탐색 스케줄러의 문제점을 볼 수 있는데, “우선도/실행시간” 값이 ‘0.20’으로 가장 작은 탐지 작업 같은 경우에는

	Surveillance	Confirmation	Acquisition	Trace tracking	Precision tracking
Priority(Value)	0.20	0.67	1.00	0.67	0.63
Processing time(Weight)					
20 random tasks / Capacity = 40					
	Trial 1	Trial 2	Trial 3		
Surveillance	0 / 5	Surveillance 0 / 8	Surveillance 0 / 9		
Confirmation	3 / 3	Confirmation 1 / 1	Confirmation 3 / 4		
Acquisition	1 / 1	Acquisition 5 / 5	Acquisition 1 / 1		
Trace tracking	2 / 3	Trace tracking 1 / 3	Trace tracking 2 / 3		
Precision tracking	2 / 8	Precision tracking 2 / 3	Precision tracking 2 / 4		
20 random tasks / Capacity = 60					
	Trial 1	Trial 2	Trial 3		
Surveillance	0 / 6	Surveillance 0 / 8	Surveillance 0 / 5		
Confirmation	7 / 7	Confirmation 0 / 1	Confirmation 5 / 6		
Acquisition	3 / 3	Acquisition 5 / 5	Acquisition 3 / 3		
Trace tracking	1 / 1	Trace tracking 2 / 2	Trace tracking 2 / 2		
Precision tracking	3 / 3	Precision tracking 4 / 4	Precision tracking 3 / 4		

그림 12. 탐색 스케줄러 시뮬레이션 결과
Fig. 12. Results of knapsack scheduler simulation.

모든 시도에서 항상 탈락하고 있다. 이렇게 되면 우선도가 낮은 작업은 계속해서 탈락하여 레이다가 제 기능을 수행하지 못할 수도 있다. 예를 들어 너무 많은 추적 작업으로 인해 탐지 작업이 오랜 시간동안 탈락한다면, 새로운 목표물을 탐지하여 새로운 추적 작업이 시작될 기회가 없어진다. 그렇게 레이다 시스템의 주체가 큰 위험에 처할 수도 있다.

3-3 제안된 스케줄러(Proposed Scheduler)

이전에 소개된 스케줄러들은 몇 가지 단점들을 가지고 있었다. TB 스케줄러는 작업의 갱신 시간보다 TB 값을 우선으로 고려하여 스케줄링하기 때문에 작업이 본래에 스케줄링되어야 하는 위치보다 한참을 지연되어 스케줄링 되었다. 이것은 작업이 적절한 시간에 실행되는 것을 방해한다. 탐색 스케줄러의 경우에는 첫 번째 단계에서 “우선도/실행시간”값이 작은 작업을 탈락시켜, 특정 작업은 한 번도 스케줄링되지 못하는 경우가 발생하였다. 그러한 문제들을 해결하기 위해 본 연구에서 제안하는 스케줄러는 일차적으로 보정된 탐색 알고리즘으로 작업들을 추려내고, TB 스케줄러에 새로운 비용함수를 적용해 작업을 나열함으로써 문제를 해결하려고 한다.

첫 번째 단계에서 탐색 스케줄러가 특정 작업을 모두 탈락시켜 버리는 경우를 방지하기 위해 작업이 단 한 번도 스케줄링되지 못한다면, 다음 스케줄링 시퀀스에서 우선도 값을 조절하여 스케줄링될 수 있도록 적응형 알고리즘을 적용하도록 한다. 스케줄링할 작업이 선택되면, 두 번째 단계에서 TB 스케줄러로 타임라인에 순서대로 선택된 작업들을 나열한다. 이때, 새로운 비용함수를 적용한다.

그림 13에서와 같이 비용함수를 생성했다. 그래프의 가로축은 시간을 의미하고, 세로축이 비용 값이다. 그래프에서 비용이 0이 되는 지점의 시간 값은 갱신 시간이고, 비용이 1이 되는 지점의 시간 값이 한계 값(limitation time)이다. 작업마다 한계 값, 갱신 값 그리고 비용함수가 주어져 있어, TB 값의 변화에 따라 비용 값도 달라진다. TB 값이 갱신 값과 같은 경우에는 비용 값이 0이 되고, TB 값이 갱신 값을 초과하여 증가하는 경우에는 비용 값이 1에 가까워진다. 비용 값이 1이 되는 것은 TB 값이 한

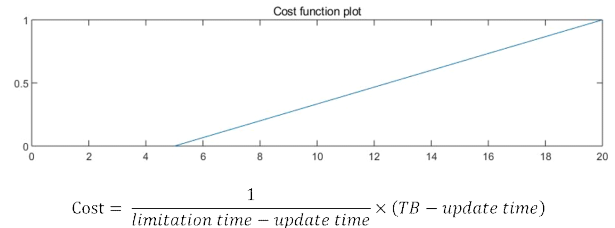


그림 13. 스케줄링 비용 함수

Fig. 13. Cost function for scheduling.

계 값에 도달하였다는 것을 의미한다. TB 값이 한계 값 혹은 그 이상으로 증가하여 1 이상의 비용 값을 가지는 작업은 즉시 스케줄링이 되어야 하는 긴급한 상황으로 간주한다.

새로운 비용함수를 이용하여 TB 스케줄러의 알고리즘을 변형해 3-1절에서 제시한 TB 스케줄러의 문제점을 해결할 수 있다. 어떠한 작업의 TB 값이 다른 작업보다 낮아 스케줄링되지 못하는 상황이라도 비용 값이 0을 넘어서면 그 우선도가 점점 증가하고, 작업의 비용 값이 1을 넘어가면 스케줄링이 긴급한 상황으로 판단하여 즉시 작업을 스케줄링할 수 있다. 따라서 이전의 TB 스케줄러처럼 어떠한 작업이 너무 오랜 시간 동안 지연되어 스케줄링 되는 경우를 방지할 수 있다.

3-3-1 제안된 스케줄러 시뮬레이션 1

제안된 스케줄러는 탐색 알고리즘으로 스케줄링할 작업을 분류하는 첫 번째 단계와 분류된 작업을 타임라인에 나열하는 두 번째 단계가 존재한다. 시뮬레이션에서 두 단계를 연속적으로 진행하기가 어려워 두 개의 서로 다른 시뮬레이션으로 나누어서 진행하였다.

시뮬레이션 1에서는 실행마다 우선도 값이 조정되는 알고리즘을 적용한 탐색 스케줄러를 이용하여 일정한 타임라인 동안 수행될 작업들을 분류할 것이다.

3-2절과의 비교를 위해 3-2절에서 사용한 작업들과 동일한 작업들을 이용하여 시뮬레이션 1을 진행하였다. 작업들이 생성될 확률도 3-2절에서 사용한 것과 동일한 마르코프 체인을 이용하였다.

각 작업의 초기 우선도 값은 그림 14에서 위에서부터 차례대로 1, 2, 3, 4 그리고 5의 값을 가진다. 그리고 스케

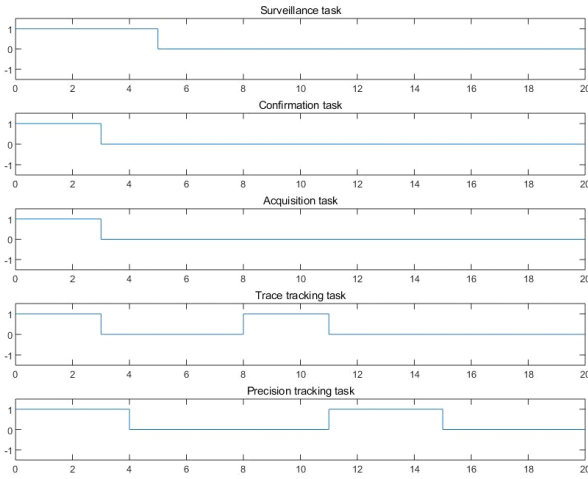


그림 14. 제안된 스케줄러 시뮬레이션 1 작업
Fig. 14. Proposed scheduler simulation 1 tasks.

줄러를 실행했을 때, 마르코프 체인을 통해 생성된 작업의 25% 이하가 스케줄링된 작업은 초기 우선도 값에 +1을 해주고, 75% 이상이 스케줄링된 작업은 초기 우선도 값에 -1을 해주는 방식으로 우선도 값을 조정하면서 3번의 실행을 하였다.

3-3-2 제안된 스케줄러 시뮬레이션 1 결과

제안된 스케줄러 시뮬레이션 1의 결과로 그림 15를 얻을 수 있었다. 3-2절의 탐색 스케줄러 시뮬레이션의 결과와는 다르게 이번 시뮬레이션에서는 trial이 반복됨에 따라 우선도 값이 작아 탈락하기만 하던 탐지 작업도 스케

20 random tasks / Capacity = 40

Trial 1		Trial 2		Trial 3	
Surveillance	0 / 9	Surveillance	4 / 12	Surveillance	1 / 8
Confirmation	3 / 3	Confirmation	0 / 5	Confirmation	2 / 2
Acquisition	1 / 1	Acquisition	0 / 1	Acquisition	7 / 7
Trace tracking	2 / 4	Trace tracking	2 / 2	Trace tracking	0 / 1
Precision tracking	2 / 3	Precision tracking	1 / 1	Precision tracking	1 / 2

20 random tasks / Capacity = 60

Trial 1		Trial 2		Trial 3	
Surveillance	0 / 4	Surveillance	5 / 9	Surveillance	1 / 8
Confirmation	5 / 6	Confirmation	0 / 4	Confirmation	3 / 3
Acquisition	5 / 5	Acquisition	3 / 3	Acquisition	5 / 5
Trace tracking	5 / 5	Trace tracking	3 / 3	Trace tracking	2 / 2
Precision tracking	0 / 1	Precision tracking	1 / 1	Precision tracking	2 / 2

그림 15. 제안된 스케줄러 시뮬레이션 1 결과
Fig. 15. Results of proposed scheduler simulation 1.

줄링되는 것을 확인할 수 있다. 3-2-2절에서 설명하였듯이 어떠한 작업이 오랜 시간동안 수행되지 못하는 것은 바람직하지 못하다. 이번 시뮬레이션 결과에서는 각 trial에서 탈락한 작업은 다음 trial에서는 우선도 값이 커져 더 많이 스케줄링되고 있음을 볼 수 있다. 제안된 스케줄러는 탈락한 작업에 우선도 보정을 하여 작업 간에 우선도 균형을 맞추고 있다는 것을 확인할 수 있다.

3-3-3 제안된 스케줄러 시뮬레이션 2

새로운 스케줄러를 위해 3-1-3절에서와 동일한 3개의 작업을 가정하였다. 그림 16에서 작업들의 실행 시간과 갱신 시간을 확인할 수 있다.

그리고 세 작업들의 비용함수는 그림 17에서 확인할 수 있다. 다음과 같은 작업으로 제안된 스케줄러의 시뮬레이션을 진행하였다.

제안된 스케줄러 시뮬레이션 2에서는 첫 번째 단계를 통해 적당하게 작업이 선택되었다고 가정하고, 비용함수를 적용함으로써 수정된 TB 알고리즘으로 작업을 타임라인에 나열하였다. 전체 레이다 타임라인의 길이는 50단위 시간으로 설정하였다.

3-3-4 제안된 스케줄러 시뮬레이션 2 결과

그림 18은 제안된 스케줄러의 결과이다. 여기서 우리는 delay 값이 현저하게 작아진 것을 확인할 수 있다. 즉, 작업들이 갱신 시간을 준수하며 스케줄링이 되는 것이다. 따라서 비용함수를 적용하면 기본적인 TB 스케줄러가 가지고 있던 문제점을 해결할 수 있다는 것을 확인할 수 있

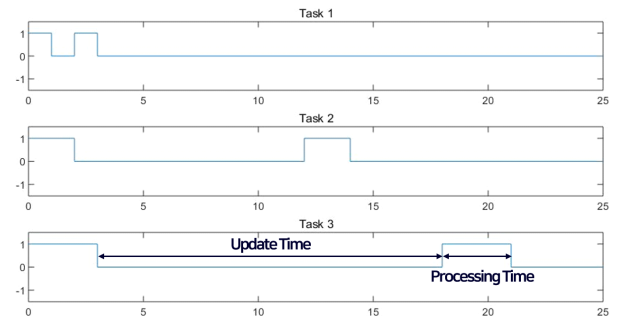


그림 16. 제안된 스케줄러 시뮬레이션 2 작업
Fig. 16. Proposed scheduler simulation 2 tasks.

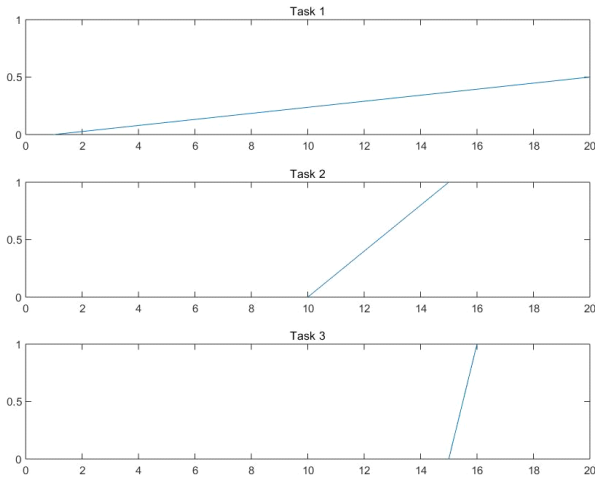


그림 17. 제안된 스케줄러 시뮬레이션 2 작업 비용함수
Fig. 17. Proposed scheduler simulation 2 task cost functions.

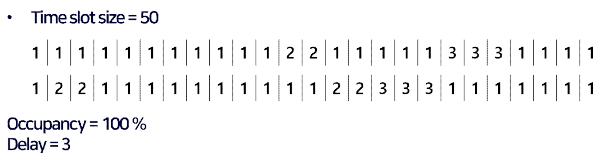


그림 18. 제안된 스케줄러 시뮬레이션 2 결과
Fig. 18. Results of proposed scheduler simulation 2.

었다. 또한 비용함수가 간단하여 기존의 TB 스케줄러와 비교해 보았을 때 계산량이 많이 늘어나지 않았다. 10번의 실행시간의 평균 시간 차이가 0.001초로 매우 작았다. 따라서 레이다 자원 중 하나인 계산(computation)의 측면에서도 손실이 없다고 할 수 있다.

IV. 결 론

다기능 레이다는 탐지, 추적 그리고 유도과 같은 다양

한 작업은 하나의 시스템에서 처리할 수 있는 유용한 레이다이다. 하지만 레이다의 시간 자원은 제한되어 있고, 그것을 효율적으로 이용하기 위해서, 스케줄링이라는 레이다 자원 관리 기술을 이용한다. 다양한 스케줄링 기법 중에 TB 스케줄러와 탐색 스케줄러를 소개하였고, 비교하여 보았다. 스케줄러들은 각각의 장점이 있지만, 단점도 존재하였다. 제안된 스케줄러를 통해 이러한 문제점을 보완하여 시간 효율적이면서도, 안정적으로 작업을 레이다 타임라인에 스케줄링할 수 있도록 하는 방법을 제시하였다.

References

- [1] Ö. Çayır, "Radar Resource Management Techniques for Multi-Function Phased Array Radars," M.S. thesis, Middle East Technical University, Ankara, Turkey, 2014.
- [2] E. Winter, P. Baptiste, "On scheduling a multifunction radar," *Aerospace Science and Technology*, vol. 11, no. 4, pp. 289-294, 2007.
- [3] M. I. Skolnik, *Introduction to Radar Systems*, Boston, Tata McGraw Hill, p. 2003.
- [4] D. Pisinger, "Algorithms for Knapsack Problems," Ph.D. dissertation, University of Copenhagen, Copenhagen, Denmark, 1995.
- [5] W. K. Stafford, "Real time control of a multifunction electronically scanned adaptive Radar(MESAR)," in *IEE Colloquium on Real-Time Management of Adaptive Radar Systems*, London, UK, 1990, pp. 7/1-7/5.
- [6] 노지은, 안창수, 김선주, "Dispatching rule에 기반한 능동 위상 배열 다기능 레이다의 빔 스케줄링 기법," *한국전자과학회논문지*, 23(1), pp. 1-13, 2012년 1월.

황 민 영 [한국과학기술원/학사과정]



2016년 3월~현재: 한국과학기술원 전기 및 전자공학부 학사과정
[주 관심분야] 레이더

신 상 진 [한화시스템/팀장]



1989년 2월: 경상대학교 전자공학과 (공학사)
1994년 3월: 경상대학교 전자공학과 (공학석사)
2004년 2월: 경상대학교 전자공학과 (공학박사)
2004년 2월~현재: 현재 한화시스템 해상

MFR 팀장

[주 관심분야] 레이더 시스템 설계, 표적 추적필터, 레이더 신호처리

양 우 용 [한화시스템/전문연구원]

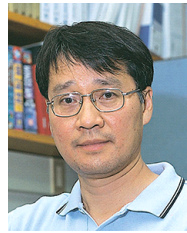


2005년 2월: 서강대학교 전자공학과 (공학사)
2007년 2월: 한국과학기술원 전기 및 전자공학부 (공학석사)
2015년 2월: 한국과학기술원 전기 및 전자공학부 (공학박사)
2007년 1월~현재: 한화시스템 레이더·

PGM 연구소 전문연구원

[주 관심분야] 레이더 시스템 설계/성능분석, 안테나 및 레이더 신호처리, 표적인식 알고리즘 설계/분석

전 주 환 [한국과학기술원/교수]



1980년: 서강대학교 전자공학과 (공학사)
1984년: 미국 Cornell University 전자공학과 (공학석사)
1989년: 미국 Stanford University 전자공학과 (공학박사)
2007년~현재: 한국과학기술원 전기 및 전자공학부 교수

[주 관심분야] 레이더