

다중 기동 표적에 대한 추적 방식의 비교

Comparison of the Tracking Methods for Multiple Maneuvering Targets

임 상 석*

Sang Seok Lim*

요 약

다중 표적의 추적은 과거 10여년 동안 레이더 응용분야에서 큰 주목을 받아 왔으며 많은 학술회의와 논문발표의 중심 과제가 되어왔다. 이 문제를 해결하기 위하여 여러 가지 추적방식들이 제시되어 왔다. 그 중 대표적인 것으로는 Nearest Neighbor (NN) 방식에 의한 비확률적인 짝배정(association) 방법과 확률적인 표적모형에 기초한 Multiple Hypothesis Test (MHT) 방식 및 Joint Probabilistic Data Association (JPDA) 방식으로 대별할 수 있다. 이러한 여러 가지 방식들은 각기 그 장점 및 단점을 가지게 되어 계산속도나 표적의 추적정확도에 있어서 큰 차이를 나타내게 된다. 본 논문에서는 NN방식, MHT 방식 및 JPDA 필터에 기초한 세 가지 추적 알고리즘을 비교하고, 시뮬레이션을 통하여 다중 기동 표적에 대하여 그 추적성능을 분석한다.

Abstract

Over last decade Multiple Target Tracking (MTT) has been the subject of numerous presentations and conferences [1979-1990]. Various approaches have been proposed to solve the problem. Representative works in the problem are Nearest Neighbor (NN) method based on non-probabilistic data association (DA), Multiple Hypothesis Test (MHT) and Joint Probabilistic Data Association (JPDA) as the probabilistic approaches. These techniques have their own advantages and limitations in computational requirements and in the tracking performances. In this paper, the three promising algorithms based on the NN standard filter, MHT and JPDA methods are presented and their performances against simulated multiple maneuvering targets are compared through numerical simulations.

I. 서 론

1-1 배 경

레이더와 같은 센서 시스템을 통해서 얻어지는 측정치는 여러 가지 요인들 (예를 들면, 다른 표적

들, 후면 잡음들, 레이더 지면 클러터 (clutter) 등) 로 인해서 잡음을 포함하게 된다. 이러한 잡음이 섞인 신호로부터 표적의 정확한 위치를 산출해 내려는 표적 추적 문제는 표적의 측정치 (measurement)로부터, 매 기준 시점마다 표적의 위치 및 속도를 추산해 내는 것이 목표이다.

이 표적 추적문제를 풀기 위해 사용되는 보편적

* 한국항공대학교 항공전자공학과 (Dept. of Avionics, Hankuk Aviation Univ.)

· 논문번호 : 97-1-4

· 접수일자 : 1997년 12월 20일

인 가정은 표적의 운동 및 그 측정치가 백색 Gaussian 잡음을 포함한다고 보는 것이다. 이러한 가설을 사용할 때 표적추적 문제의 해는 칼만 (Kalman) 필터나 또는 다른 복잡한 비 선형필터에 의하여 얻어지게 되고, 가정된 잡음 모델과 실제 잡음의 차이는 결국 추산되는 표적의 위치 및 속도에 오차를 유발하게 된다.

측정치의 근원이 확실한지 아닌지에 대해서도 또한 가지 부수적인 중요한 문제가 수반되는데, 측정된 데이터가 실제로 문제의 표적에서 수신된 것인지 아니면 다른 표적과의 간섭에 의한 것인지 또는 아예 클러터인지를 확실히 알 수 없기 때문이다. 이러한 표적측정치의 근원에 대한 문제는 data association (DA: 짝배정) 문제로 불리며, 실제표적과 추산표적 또는 트랙 (track)과의 짝짓기(pairing)를 해 주는 방식이다. 명백히, 이러한 상황에서는 N개의 표적에 대한 추적의 경우 계산절차는 한 개의 표적에 대한 계산의 N배의 노력과 비교되지 않을 만큼 복잡하고 어렵게 된다. 그 이유는 산출된 표적과 측정치와 일대일로 짝을 짓는 것은 결코 단순한 작업이 아니기 때문이다. Bar-Shalom과 Fortmann[1] 은 그의 저서에서 다중표적과 DA를 위한 수학적인 기초자료를 발표하였다. 최근에 발표된 논문집[2],[15] 에서는 여러가지 적용사례와 DA방식에 대해서 새로운 개선결과를 발표하였다.

DA 문제를 다루는 여러 가지 방식들이 제안되었지만 결국 대별하면 두 가지로 요약할 수 있다. 그 첫째는 비 확률적인 모델로서, 가장 적합하다고 생각되는 측정치를 실제 측정치인 양으로 간주하여 처리하는 DA방식이다. 이러한 방식의 대표적인 것은 NNSF (NN Standard Filter) 및 TSF (Track Splitting Filter)이다. 또 다른 방식은 확률적인 모델로서, 짝짓기 확률을 Bayes의 공식에 입각해서 계산하고 이것을 표적위치 추산에 반영하는 것이다. 그 대표적인 예로는 MHT와 JPDA 방식이 있다.

본 논문에서는 비 확률적인 방법의 NNSF와 확률적인 방식의 MHT 및 JPDA를 비교하고 그 장단점을 시뮬레이션을 통해서 비교 해석한다. NN 표준필터는 각 추적 시점에서 바로 한 단계 전 시점

에서의 짝짓기와 현재 시점에서의 측정치를 근거하여 단 한 개의 확고한 짝짓기 해를 산출해 낸다. 그러나 이러한 방식의 해는 실제로는 최선의 해가 아닐 수도 있는데 그 이유는 이 NN방식의 해는 결국 이전에 습득한 모든 측정자료들은 완전히 이용하지 않았기 때문이다. 따라서 일단 부정확한 짝짓기가 이루어지면, 이로 인해서 야기되는 추적상의 오차는 끝내 잘 교정되어 지지 못하기 때문이다.

MHT 방식은 여러 가지 (정확히 말해서 무수히 많은) 가능한 모든 짝짓기 해를 계속 유지하고 있으면서, 불확실하게 간주되는 측정값의 총 내력을 참고로 사용해서 전혀 일어날 것 같지 않은 짝짓기를 제거함으로써 최상의 선택을 얻어내는 방법이다. 이러한 접근방식은 최상의 해를 산출해 줄 수는 있으나 일반적으로 말해서 불확실한 추적 주기를 갖게 된다. 가장 큰 약점은 추적해야 할 표적이 한 곳에 밀집해 있는 경우 그 짝짓기 수는 추적 시간이 증가함에 따라서 기하 급수적으로 증가하기 때문이다. 이것은 또 컴퓨터 계산시간이 많이 소모되어 큰 부담을 주게 된다는 것을 의미한다. 따라서 MHT 방식의 응용에는 이 짝짓기 수를 줄여서 표적가설 (track hypothesis)의 수를 줄이는 것이 최대의 관심사가 된다. 또 다른 약점은 흔히 이 짝짓기에 의한 표적의 결정은 실제의 측정치를 얻은 시각에서 이루어지지 않고 늘 여러 스캔(Scan) 뒤에 가서야 비로소 결정된다는 사실이다. 이것은 실제의 표적 추적 상황에서는 상당히 불리한 조건으로 간주될 수도 있기 때문이다.

JPDA 방법은 짝짓기에 있어서 단정적으로 결정하지는 않는다. 대신에 모든 가능한 짝짓기 후보에 대해서 적절히 계산된 확률을 곱해서 합산함으로써 최선의 표적의 추산치를 계산해 내는 방식이다. 이것은 어디까지나 표적의 추적 값을 계산할 때 부정확한 측정치에는 작은 값의 확률을 사용하고, 가능성이 높은 실제표적의 측정치에는 높은 확률을 적용했을 때만 바른 결과를 산출해 준다. 이 방식은 매 추적 시점마다 기다림이나 지연 없이 즉시 단 한 개의 추적 추산치를 계산해 준다는 장점이 있다. 반면에 이러한 불확실한 짝짓기를 실제 표적의 계산에 사용함으로써 그 정확도가 떨어지고 잘못된 짝

깃기와 오차의 누적으로 결국 추적필터가 발산할 가능성도 있다. 더욱이 JPDA 방식은 초기에 표적의 초기화 단계에 있어서 확실한 개시 방안을 제공해 주지 못한다.

본 논문에서는 많은 추적 방식 중에서 대표적인 NNSF, MHT 및 JPDA 방식에 기초한 추적알고리즘을 실제로 직접 활용할 수 있을 정도로 상세하게 기술하고 그 추적성능을 시뮬레이션을 통해서 비교 분석한다. 다음 절에서는 먼저 알고리즘을 기술하기 위한 기본적으로 사용하는 표적과 측정치의 모델을 언급한다. 제 2절에서는 NN DA 방식을 이용한 알고리즘을 제시하고, 제3절에서는 수정된 MHT 알고리즘을 발표한다. JPDA 알고리즘은 제 4절에서 소개한다. 이들 세 가지 알고리즘의 추적 특성을 다중 기동표적에 대하여 시뮬레이션한 결과를 제 5절에서 비교하고 해석한다.

1-2 표적과 측정치

추적해야 할 어떤 물체에 대해서, 그 운동방식은 비연속적인 모델을 사용할 경우 다음과 같이 나타낼 수 있다.

$$x(k+1) = F(k)x(k) + G(k)u(k) \quad (1)$$

여기서 $x(k)$ 는 n 차의 상태 벡터로서 추적할 물체이고, $F(k)$ 는 천이행렬, 그리고 $u(k)$ 는 m 차 상태 벡터로서 표적의 기동이나 모델오차를 나타내는데 일반적으로는 평균이 0이고 분산(variance)이 $Q(k)$ 인 백색 Gaussian으로 가정한다. TWS (Track-While-Scan) 시스템에서는 k 번째 샘플은 대략 시점 kT (이때 T 는 센서의 스캔(scan) 간격) 부근에서 주어지게 된다. 추적하려는 물체의 센서 측정치를 나타내기 위해서 다음과 같은 관측 방정식을 사용한다.

$$y(k) = H(k)x(k) + v(k) \quad (2)$$

여기서 $y(k)$ 는 m 차 센서 측정 벡터이고 때로는 센서 리포트(reports)라고 불리기도 한다. $v(k)$ 는

관측잡음을 나타내는 백색 Gaussian 벡터로서 평균이 0이고 분산 $R(k)$ 를 갖는다. 열에 의한 FA(false alarm), 클러터(clutter) 또는 기타 허위 표적으로부터 발생하는 잡음 섞인 센서 리포트를 나타내는 관측 방정식은

$$y(k) = H(k)\hat{x}(k, k-1) + w(k) \quad (3)$$

로 가정되며, 여기서 $w(k)$ 는 백색이고 어떤 체적 V 에 균일하게 분포된 잡음이며 그 중심은 예측측정치로서

$$\hat{y}(k, k-1) = H(k)\hat{x}(k, k-1) \quad (4)$$

로 주어진다. 이러한 잡음을 수반한 체적 V 내의 리포트들은 평균 βV (여기서 β 는 정상화되지 않은 리포트의 밀도)를 갖는 Poisson 분포를 갖는다. 이러한 표적과 측정치에 대하여 추적필터는 scan k 까지의 측정치를 사용해서 상태 추산값 $\hat{x}(k, k)$ 및 한 scan 예측 추산치 $\hat{x}(k+1, k)$ 를 계산해 준다. 이때 사용되는 필터 방정식 및 DA를 위한 알고리즘을 다음의 제 2절, 제 3절 및 제 4절에서 기술하고 그 장단점을 비교한다. 제 5절에서는 이들 알고리즘을 사용했을 때의 다중 기동표적에 대한 추적결과를 비교 해석한다.

II. NNSF 알고리즘

2-1 NNSF 알고리즘

- 1단계 : scan 지수 k 를 1로 입력하고, 기타 필요한 모든 변수들을 입력한다.
- 2단계 : 시스템 변수 F, Q, H, R 을 입력한다.
- 3단계 : 표적의 궤적을 시뮬레이션한다.
- 4단계 : 측정치를 시뮬레이션한다. (추적문제에 실제로 적용하는 경우에는 센서로부터 측정 데이터를 받아들인다.)
- 5단계 : 측정 데이터를 짝배정에 사용할 것인지 아닌지 판별한다. 만일 측정치가 다음의 조건을 만족하면 유효한 측정치로 판정

하고 아니면 무효한 측정치로 간주하여 DA에서 제외해 버린다.

로 되돌아 간다.

$$\begin{aligned} & \{\hat{y}(k)\}' S^{-1}(k) \{\hat{y}(k)\} \leq g^2 \\ & (\text{"g-}\sigma\text{" 타원게이트}) \\ & \hat{y}(k) = y(k) - \hat{y}(k, k-1) \end{aligned} \quad (5)$$

6단계 : 짝배정 (Assignment) 행렬을 만든다. 이 행렬의 각 요소들은 N_{ob} 개의 관측치를 N_r 개의 트랙 각각에 배정할 때 계산한 정규화 된 거리함수 (normalized distance function) 값을 나타낸다. 만일 위의 식(5)에 주어진 타원 게이트 판별에 합격하지 못한 트랙-관측치 짝에는 대단히 큰 거리함수값을 사용하면 된다

7단계 : 이 짝배정 행렬을 푼다. 이 행렬에서 최선의 해는 정규화된 거리함수 값의 합이 최소인 경우이며, 다음에 기술하는 수정된 Munkres 최적배정 알고리즘을 사용하여 구할 수 있다.

8단계 : 최적 배정 행렬의 해에 따라서 각 측정치와 트랙들을 짝 짓는다.

9단계 : 다음의 필터를 사용해서 각 추산치 및 예측치를 계산한다.

$$\begin{aligned} A(k) &= P(k, k-1)H'(k)[H(k)P(k, k-1) \\ & \quad H'(k)+R(k)]^{-1} \end{aligned} \quad (6)$$

$$\hat{x}(k, k) = \hat{x}(k, k-1) + A(k)[y(k) - H(k)\hat{x}(k, k-1)] \quad (7)$$

$$P(k, k) = [I - A(k)H(k)]P(k, k-1) \quad (8)$$

$$\hat{x}(k+1, k) = F(k)x(k, k) \quad (9)$$

$$\begin{aligned} P(k+1, k) &= F(k)P(k, k)F'(k) + \\ & \quad G(k)Q(k)G'(k) \end{aligned} \quad (10)$$

10단계 : 만일 $k=k_f$ (추적임무 완료)이면 끝낸다. 아니면, $k=k+1$ 로 놓고, 제 4단계

2-2 수정된 Munkres 최적배정 알고리즘

다음의 알고리즘은 Burgeois와 Lassalle^[12]에 의해서 수정된 Munkres 최적배정 알고리즘이다. 이 방식은 실제 응용에 있어서 편리한 점이 있는데 행렬의 행과 열의 수가 같지 않아도 된다는 점이다. 이 알고리즘을 쉽게 기술하기 위해서, 행렬의 행과 열에 대해 특별히 표식을 붙일 수 있는데 이것을 "커버됐다"는 표현을 쓰기로 한다.

최적배정 알고리즘

1단계 : 처음에는 어떤 행이나 열도 커버되지 않고, 0 들은 "*"나 "" 표식을 붙이지 않는다.

2단계 : 행이나 열의 수 중 작은 것을 v 라 한다.

3단계 : 만일 열의 수가 행의 수보다 크면 제 6단계로 간다.

4단계 : 배정 행렬의 각 행에 대해서, 가장 작은 요소 값을 해당 행마다 뺀다.

5단계 : 만일 열의 수가 행의 수보다 크면 제 7단계로 간다.

6단계 : 배정 행렬의 각 열에 대해서, 각 열의 가장 작은 요소 값을 해당 열마다 뺀다. 그런 다음 제 7단계를 수행한다.

7단계 :

1) 배정 행렬에서 한 개의 "영" ("Z")를 찾는다.
2) 만일 "*"를 붙인 Z가 발견되지 않으면, 그 해당 행이나 열의 Z에 "*"를 붙인다. (즉, Z*로 한다). 이 행렬의 모든 영("Z")에 대해서 이 과정을 반복한다.

8단계:

1) Z*를 포함하는 각 열을 "커버"한다.
2) 만일 v 개의 열이 커버되면, Z*의 위치가 행-열 짝짓기 (즉, 관측치-track의 쌍)를 형성한다. 이로써 이 알고리즘은 끝난다. 만일 v 개의 열이 커버되지 않았으면 다음 9단계를 수행한다.

9단계:

- 1) 한 개의 커버 않된 “영”을 선택하고 “*”을 붙인다. (즉 “Z’”)
- 2) 만일 Z’가 있는 행에 Z*가 있으면 제 10단계로 간다.
- 3) 만일 Z’의 행에 Z*가 있으면, 이 행을 커버하고 Z*의 열을 커버 해제한다.
- 4) 모든 “영”이 다 커버될 때까지 이 과정을 반복하고 나서 제 11단계로 간다.

10단계:

- 1) Z_0 는 커버 해제된 Z’을 나타낸다 하자. Z_0 의 열 안에 Z*가 없으면 제 10.6단계로 간다.
- 2) Z_1 은 Z_0 의 열 안에 있는 Z*를 나타낸다고 한다.
- 3) Z_2 는 Z_1 의 행 안에 있는 Z’을 나타낸다고 한다.
- 4) 제 10.2 단계와 제 10.3 단계를 반복해서 한 Z_2 의 열 안에 Z*가 모두 없어질 때까지 계속한다.
- 5) 이 과정에서 각 Z*를 해제해서 Z로 한다.
- 6) 이 과정에서 각 Z’을 모두 Z*로 한다.
- 7) Z’은 모두 “*”을 제거해서 Z로 한다. 그리고 각 커버된 줄들을 해제한다.
- 8) 제 8단계로 간다.

11단계:

- 1) 배정 행 열의 요소들 중에서 커버되지 않은 최소의 요소를 찾아서 그 요소를 “m”라 하자. m은 양수이다.
- 2) 이 m값을 각 커버된 행마다 더해준다.
- 3) 커버되지 않은 열에 대해서는 m 값을 빼준다.
- 4) “*”붙은 것, “*” 붙은 것 또는 커버 않된 줄들은 변경하지 말고 제 9단계로 간다.

Ⅲ. 다중 표적의 추적을 위한 수정 MHT 알고리즘

3-1 수정 MHT 흐름선도

수정 MHT(MMHT) 알고리즘의 주요 기능은 Clustering, 가설설정 (hypothesis generation)

가설 간소화 (Pruning) 및 트랙관리 등으로 구성된다. 서브루틴 CLUST는 새로운 측정치 데이터를 이전 scan에서 결정한 클러스터 (cluster)와 짝짓는 과정이다. 클러스터란 다른 클러스터 안에 있는 다른 트랙가설들과 전혀 연관되지 않는 트랙가설들만의 집단이다. 한 개의 클러스터 안에 있는 트랙가설들은 다른 클러스터 안에 있는 트랙들 또는 측정치를 공유하거나 사용하지 않는다. 클러스터 트랙가설들을 작은 여러 개의 소 클러스터로 나누는 가장 큰 이유는 복잡한 추적 문제를 각각 독립적으로 취급할 수 있는 간단한 여러 개의 추적 문제로 나누어서 좀더 간편하게 다루려는 것이다. MMHT방식에 의한 추적 알고리즘은 그림 1과 같다. 가설설정 서브루틴 HGEN은 각 클러스터 별로 gating 판정을 거친 측정 데이터를 기존의 트랙 또는 가설들에 배정해서 새로운 가설들을 만들어 낸다. 이 새로운 가설들에 대해서 가설확률 (hypothesis probability)을 PROB 서브루틴에서 계산해 낸다. 클러

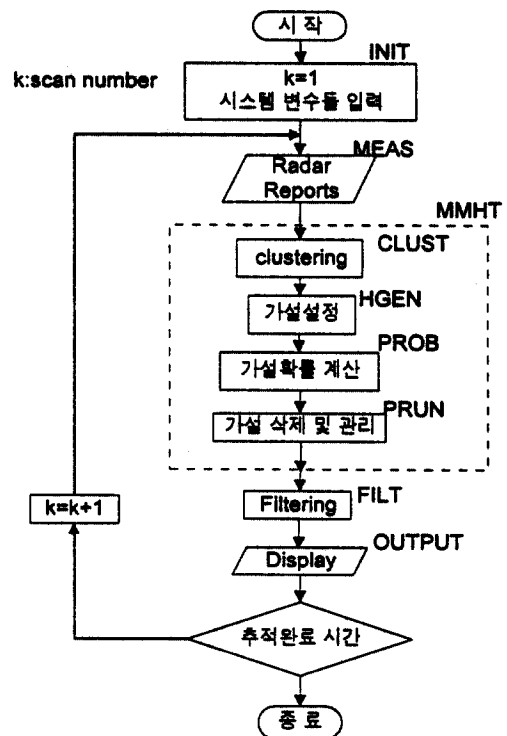


그림 1. 수정 MHT 알고리즘의 흐름선도

Fig. 1. The flow chart of the MMHT algorithm.

스터 서브루틴과 가설발생 서브루틴 모두 다 가설 간소화 방식을 사용해서 기하급수적으로 증가하는 트랙가설들의 수효를 줄인다. 서브루틴 PRUN에서는 어떤 판정기준 (예를 들면 가설확률)을 만족하는 가설들은 남겨두고, 만족스럽지 못한 가설들은 모두 지워 버린다. 수정된 MHT 방식에서는 매 N-scan 마다 각 클러스터별로 최선의 트랙가정 한 개만 남기고 나머지는 모두 제거해 버리는 방법을 사용한다. 필터 서브루틴 FILT는 위의 개요에서 설명한 필터를 사용해서 각 트랙에 대한 추산치와 예측치를 계산해 낸다. MMHT와 Reid가 제안한 MHT 방식은 트랙가설을 단순화시키는 전략에 있어서 차이가 있는데, MMHT는 DTH (Dynamic thresholding) 및 N-scan Pruning을 함께 사용해서 MHT방식에서는 급격히 증가하는 트랙가설의 수를 MMHT에서는 보다 더 빠른 속도로 감소시킬 수 있다는 점이다.

3-2 수정 MHT 알고리즘

1단계: 스캔변수 k 를 1로 입력하고 기타 제어변수들을 입력한다.

2단계: 시스템 변수 F, Q, H, R 을 입력한다.

3단계: 표적들의 궤적을 시뮬레이션 한다.

4단계: 측정치를 시뮬레이션 한다. (추적문제에 실제로 적용하는 경우에는 센서로부터 측정 데이터를 받아들인다.)

5단계: 각 트랙가설 $h_l, 1 \leq l \leq L(k-1)$,에 대해서 무조건적인 추산변수인 $P_l(k, k-1), K_l(k-1), \hat{x}(k, k-1), P_r(k, k-1), P_r(k-1)$ 를 받아들인다.

6단계: $\hat{y}(k, k-1)$ 를 중심으로 하는 판정게이트를 설정하고 필터에 사용될 N_k 개의 센서 리포트를 선정한다. 측정데이터는 식(5)에 주어진 타원게이트를 사용해서 유효한 지를 판정한다.

7단계:

1) 새로운 트랙가설 h_l 을 $1 \leq l \leq (1+N_k) \cdot L(k-1)$ 에 대해서 식별해 낸다.

2) 가설 h_l 의 확률 $P_r(k)$ 를 주어진 측정치 $Y^k =$

$\{Y(1), Y(2), \dots, Y(k)\}$ 에 대해서 다음과 같이 계산한다.

$$\begin{aligned} P_r^l(k) &\equiv P_r(h_l(k) | Y^k) \\ &= \frac{1}{c} P_D^{N_{DT}} (1 - P_D)^{(N_{DT} - N_{DT})} \beta_{FT}^{N_{FT}} \beta_{NT}^{N_{NT}} \cdot \\ &\quad \left[\prod_{m=1}^{N_{DT}} N(y_m - H\hat{x}, S) \right] P_r^l(k-1) \end{aligned} \quad (11)$$

여기서 $P_D, \beta_{FT}, \beta_{NT}$ 는 각각 표적검출 확률, 허위 표적 밀도, 그리고 새표적밀도를 나타낸다. c 값은 정상화하기 위한 상수이고 $P_r^l(k-1)$ 은 가설 $h_l(k-1)$ 의 확률이다. 또 R 은 정구분포로서

$$\begin{aligned} N(x, S) &\equiv \exp(-0.5x'S^{-1}x / \sqrt{(2\pi)^n |S|}), \\ S &= HFH' + R \end{aligned} \quad (12)$$

로 주어진다. 위의 식(12)에서 \bar{P} 는 스캔($k-1$)에서의 가설 $h_l(k-1)$ 에 대한 표적 추산치의 오차분산이고 R 은 측정잡음의 분산이다.

8단계: 각 가설 $h_l(k)$ 에 대해서

$$\begin{aligned} P_h(k, k-1) &= P_l(k, k-1) - P_l(k, k-1) \\ &\quad \cdot H'(k)S^{-1}(k)H(k)P_l(k, k-1) \end{aligned} \quad (13)$$

$$\begin{aligned} K_h(k, k-1) &= K_l(k, k-1) + P_l(k, k-1) \cdot \\ &\quad H'(k)S^{-1}(k, k-1)[y(k) - \\ &\quad \hat{y}(k, k-1) - H(k)K_l(k, k-1)] \end{aligned} \quad (14)$$

를 계산한다. 또는 게이트 안에 아무 측정 데이터도 수신되지 않으면 다음 식을 사용한다.

$$K_h(k, k-1) = K_l(k, k-1) \quad (15)$$

$$P_h(k, k-1) = P_l(k, k-1) \quad (16)$$

9단계: 각 가설 h_l 에 대해서 다음을 계산한다.

$$A(k) = \sum_{h=1}^{L(k)} [P_h^l(k)K_h(k, k-1)] \quad (17)$$

$$P_h(k,k) = P_h(k,k-1) \quad (18)$$

$$K_h(k,k) = K_h(k,k-1) - A(k) \quad (19)$$

10단계: 새로운 최적의 추산치를 계산한다:

$$\hat{x}(k,k) = \hat{x}(k,k-1) + A(k) \quad (20)$$

오차 분산을 계산한다:

$$P(k,k) = \sum_{h=1}^{L(k)} [P_h^h(k)\{P_h(k,k-1) + K_h(k,k-1) K_l(k,k-1)\}] - A(k)A'(k) \quad (21)$$

11단계: 다음 scan을 위해서 예측치를 다음과 같이 계산한다.

$$K_h(k+1, k) = F(k)K_h(k,k) \quad (22)$$

$$P_h(k+1, k) = F(k)P_h(k,k)F'(k) + G(k)Q(k)G'(k) \quad (23)$$

$$\hat{x}(k+1, k) = F(k)\hat{x}(k,k) \quad (24)$$

$$P(k+1, k) = F(k)P(k,k)F'(k) + G(k)Q(k)G'(k) \quad (25)$$

12단계: 만일 $k=k_f$ (추적완료)이면 끝낸다. 아니면, $k=k+1$ 로 하여 제 4단계로 간다.

주1 : 위의 식 (11)은 Reid[10]에 의해서 유도되었다. 이때 사용된 기본 가정은 부피가 C 인 추적 공간 안에 수신되는 리포트의 수는 평균 bC , (b 는 정상화된 리포트 밀도)인 Poisson분포를 갖는다는 것이다. 또 식(11)은 레이이다와 같은 제 1종 (type 1) 센서에만 사용해야 하고, 만일 제 2종 (type 2) 센서가 사용되면 참고문헌 [10]에 제시된 것처럼 수정이 되어야 한다. 다른 형태의 가설확률 계산식이 Bar-Shalom [1]에 의해 제안되었다.

IV. JPDA 알고리즘

JPDA 알고리즘을 기술하면 다음과 같다.

1단계: 스캔변수 k 를 1로 입력하고 기타 제어변수들을 입력한다.

2단계: 시스템 변수 F, Q, H, R 을 입력한다.

3단계: 표적들의 궤적을 시뮬레이션 한다.

4단계: 측정치를 시뮬레이션 한다. (추적문제에 실제로 적용하는 경우에는 센서로부터 측정 데이터를 받아들인다.)

5단계: $y'(k,k-1)$ 를 중심으로 하는 판정게이트를 설정하고 필터에 사용될 센서 리포트를 선정한다. 측정데이터는 식(5)에 주어진 타원게이트를 사용해서 유효한지를 판정한다. 사건 (event) η 에 대해서 유효배정 행렬 $\Omega(\eta)$ 를 다음과 같이 구성한다.

$$\Omega(\eta) = \{\omega(j, t; \eta) : j = 1, 2, \dots, N_m; t = 1, 3, \dots, N_j\} \quad (26)$$

여기서 사건 η 가 일어나면 (즉, 측정치 j 가 표적 t 의 판정 게이트 안에 포함되면),

$$\omega(j, t; \eta) = 1 \text{이고, 아니면 } \omega(j, t; \eta) = 0 \text{이다.}$$

6단계: 각각의 표적 $t=1, 2, \dots, N_t$ 에 대해서 측정치 j 까지의 상대거리 (residual)를 계산한다.

$$\tilde{y}'_j(k) = y_j(k) - \hat{y}'(k) \quad (27)$$

여기서 $\hat{y}'(k) = H(k)\hat{x}(k, k-1)$ 이다.

7단계: 행렬 $\tilde{y}'_j(k)$ 에 대한 공분산 행렬을 다음과 같이 계산한다.

$$S'_j(k) = H(k)P(k, k-1)H'(k) + R(k) \quad (28)$$

8단계: Bayes의 공식을 사용하여 사건 η 의 확률 $P(\eta)$ 을 다음과 같이 계산한다.

$$\begin{aligned}
 P_r(\eta) &= \text{prob}\{\eta|Y(k)\} \\
 &= (\lambda^e/c) \prod_{j=1}^{N_m} \left[\frac{-0.5(\tilde{y}_j')' S_j^{-1} \tilde{y}_j}{\sqrt{2\pi N_m |S_j|}} \right]^{\tau_j} \\
 &\quad \prod_{j=1}^{N_l} (1-P_b^j)^{1-\delta_j} (P_b^j)^{\delta_j} \quad (29)
 \end{aligned}$$

여기서 S^{-1} 은 $S_j'(k)$ 의 역행렬이다.

9단계: 각각의 표적 $t=1, 2, \dots, N_t$ 에 대해서 다음을 계산한다.

$$\theta_j^t(k) = \sum_{i=1}^{N_o} [P_r(j) \omega(j, t; i)] \quad (30)$$

또 $t=0$ 에 대하여

$$\theta_j^0(k) = 1 - \sum_{j=1}^{N_m} [\theta_j^t(k)] \quad (31)$$

을 계산한다.

10단계: 각각의 표적 $t=1, 2, \dots, N_t$ 에 대해서 다음을 계산한다.

$$\tilde{y}^t(k) = \sum_{j=1}^{N_m} [\theta_j^t(k) \tilde{y}_j^t(k)] \quad (32)$$

11단계: 필터 이득 $A(k)$ 를 계산한다.

$$A(k) = P(k, k-1) H'(k) S^{-1}(k) \quad (33)$$

12단계: $t=1, 2, \dots, N_t$ 에 대하여 상태변수와 공분산 행렬을 갱신한다.

$$\hat{x}(k, k) = \hat{x}(k, k-1) + A(k) \tilde{y}^t(k) \quad (34)$$

$$P(k, k) = P(k, k-1) - A(k) S_j'(k) A'(k) \quad (35)$$

13단계: 상태변수와 공분산 행렬의 예측치를 계산한다.

$$\hat{x}(k+1, k) = F(k) \hat{x}(k, k) \quad (36)$$

$$\begin{aligned}
 P(k+1, k) &= F(k) P(k, k) F'(k) \\
 &\quad + G(k) Q(k) G'(k) \quad (37)
 \end{aligned}$$

14단계: 만일 $k=k_r$ 이면, 추적을 끝낸다. 아니면 $k=k+1$ 로 갱신하고 4단계로 돌아간다.

주2: 위에서는 연합사건 확률을 계산할 때 식 (30)을 사용하였다. 이 식은 허위 측정치의 확률 질량(mass) 함수가 식(38)과 같이 Poisson 분포를 갖는다는 가정에서 유도된 것이다 (참고문헌 [1]의 방정식 (9.42) 참조).

$$\mu(\epsilon) = \exp(-\lambda V) \frac{\lambda V^\epsilon}{\epsilon!} \quad (38)$$

이 식에서 λ 는 허위 측정치의 밀도이고, $V(k)$ 는 판정영역의 부피이다. 따라서 $\lambda V(k)$ 는 문제의 게이트 안에 있는 허위 측정치의 평균값이다. 만일 확률 밀도함수가 균일분포 (uniform distribution)이면 적절한 다른 식(참고문헌 [1]의 (9.46)식)을 사용해서 사건의 확률을 계산해야 한다.

V. NN, 수정 MHT 및 JPDA 알고리즘의 추적 성능 비교

여기서는 NN표준필터, 수정 MHT 및 JPDA방식에 의한 추적성능을 시뮬레이션을 통하여 다중기동표적에 대해서 비교한다.

위의 세 절에서 열거한 세 가지 알고리즘을 C언어를 사용하여 프로그래밍하고 시뮬레이션을 수행하였다. 측정치는 그림 2에 나타난 바와 같이 아홉개의 무리로 형성된 몇 개의 다중표적에 대한 레이더 리포트이다. 그림 2에서 6군의 표적은 직선으로 운동하고, 3군의 표적은 기동표적으로서 반경이 점차로 커지는 나선형 궤적을 가진다. 이들 표적들은 여러 번 서로 교차하기도 하며 측정 리포트들이 스캔로스(scan loss)를 포함하고 4군의 표적들은 매

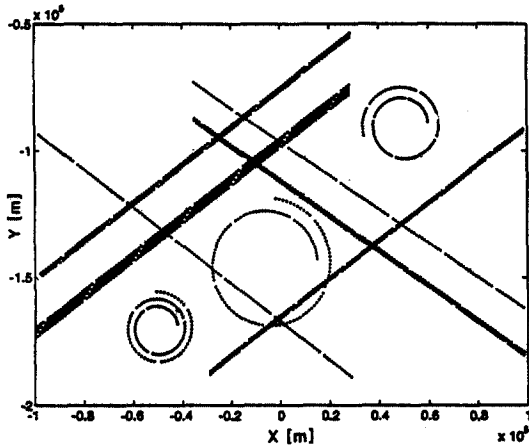


그림 2 측정 데이터
Fig. 2. The measurement data.

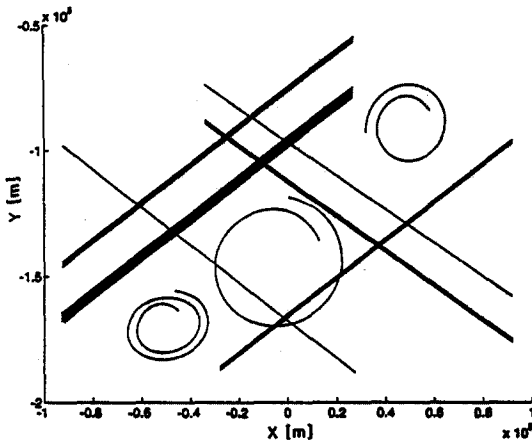


그림 3 NN 알고리즘에 의한 추적 결과
Fig. 3. Tracking results using the NN algorithm.

우 근접해 비행하는 트랙이다.

NN 표준필터에 의한 추적 결과는 그림 3과 같다. 이 결과를 보면 NN 방식을 사용하면 기동 표적에 대하여 추적 신뢰도가 떨어지는 것을 알 수 있으며 컴퓨터 사용시간은 상당히 양호한 것으로 판명되었다.

그림 4는 수정 MHT 방식에 의한 추적결과를 나타낸다. 이 결과를 보면 여러 가지 형태의 추적하기 어려운 표적에 대하여 신뢰도 높은 추적 결과를 보여준다.

그림 5는 JPDA 방식을 사용했을 때의 추적결과를 나타낸다. 이 결과에 의하면 JPDA는 직선트랙에 대한 추적 성능은 NN이나 수정 MHT와 유사하지만 기동표적이나 스캔로스와 같은 경우 상당히 왜곡된 트랙을 산출하는 것을 알 수 있다.

특히 교차하는 표적이나 스캔로스와 같은 다루기 어려운 측정 데이터에 대해서도 수정 MHT는 한결 같이 정확한 트랙을 산출해 준다. 반면에 NN 방식

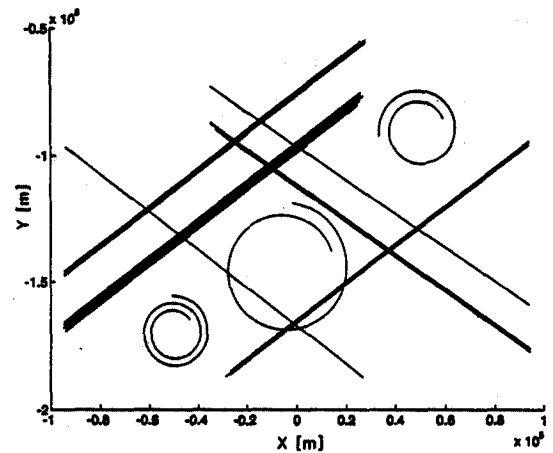


그림 4 수정 MHT 알고리즘에 의한 추적 결과
Fig. 4. Tracking results using the MMHT algorithm.

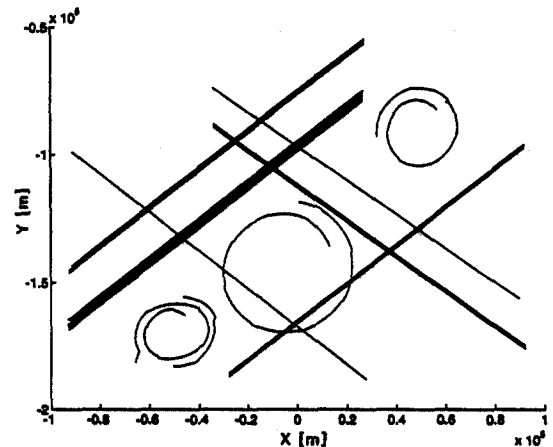


그림 5 JPDA 알고리즘에 의한 추적 결과
Fig. 5. Tracking results using the JPDA algorithm.

이나 JPDA 알고리즘을 사용한 결과(그림 3과 그림 5)는 기동표적에 대하여 정확한 트랙을 산출하지 못하며 경우에 따라서는 트랙을 잃어버리거나 완전히 다른 엉뚱한 트랙을 보여 주기도 한다.

수정 MHT 방식은 신뢰도 높은 추적성능을 보여주지만 컴퓨터 사용 시간이 NN 표준필터의 경우, 보다 길고 메모리 또한 많이 요구되는 약점이 있다. 이 시뮬레이션에 의하면 수정 MHT 알고리즘의 경우 사용된 메모리는 NN알고리즘의 5배, 소요된 CPU 시간은 대략 4배 정도였다. 또 JPDA 알고리즘의 경우는 NN방식에 비하여 메모리 사용은 3배, 소요된 CPU 시간은 2.5배 정도로 나타났다. 그러나 세 알고리즘 모두 다 실시간(real time) 응용에는 충분할 만큼 빠른 알고리즘으로 확인되었다.

VI. 결 론

본 논문에서는 NN 표준필터, 수정 MHT 및 JPDA 방식에 의한 추적 알고리즘을 단계별로 기술하였다. 이 세 알고리즘은 실제로 추적문제에 적용하기 용이하도록 알고리즘 형태로 상세히 기술하였다. 또 이 세 알고리즘을 시뮬레이션을 통하여 다중 기동표적의 추적에 적용하여 그 성능을 예증하였다. 예측했던 바와 같이 NN표준 필터와 JPDA 방식은 적은 메모리를 사용하고 계산속도가 빠르지만 추적 정확도는 수정 MHT방식보다 떨어진다. 수정 MHT는 추적신뢰도가 높은 반면에 큰 메모리를 사용해야 하고 계산속도가 NN알고리즘이나 JPDA 방식에 비하여 상당히 더 느리다. 이 결과로부터 세 가지 방식의 장단점을 분명히 확인하였으며, 아직도 보다 빠르고 신뢰도 높은 효율적인 추적 알고리즘 개발이 요청되고 있다.

참 고 문 헌

- [1] Y. Bar-Shalom, and T. E. Fortmann, *Tracking and Data Association*, Academic Press, New York, 1988.
- [2] Y. Bar-Shalom(Ed.), *Multitarget-Multisensor Tracking: Advanced Applications*, Artech House, Norwood, MA, 1990.
- [3] S. S. Blackmann, *Multiple Target Tracking with Radar Applications*, Artech House, Dedham, MA, 1986.
- [4] Y. Bar-Shalom, "Tracking Methods in a Multitarget Environment", *IEEE Trans. Automat. Contr.*, vol. 23, pp.618-626, Aug. 1978.
- [5] R. W. Settler, "An Optimal Data Association Problem in Surveillance Theory", *IEEE Trans. Military Electron.*, vol. MIL-8, pp. 125-139, Apr. 1964.
- [6] R. A. Morefield, "Application of 0-1 Integer Programming to Multitarget Tracking Problems", *IEEE Trans. Automat. Contr.*, vol. AC-22, pp.302-312, June 1977.
- [7] R. A. Singer, and R. G. Sea, "New Results in Optimizing Surveillance System Tracking and Data Correlation Performance in Dense Multitarget Environments", *IEEE Trans. Automat. Contr.*, vol. AC-18, pp. 571-581, Dec. 1973.
- [8] R. A. Singer, R. G. Sea, and K. Hou-sewright, "Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environment", *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 423-432, July 1974.
- [9] Y. Bar-Shalom, and E. Tse, "Tracking in a Cluttered Environment with Probabilistic Data Association", *Automatica*, vol. 11, pp. 451-460, Sept. 1975.
- [10] D. B. Reid, "An Algorithm for Tracking Multiple Targets", *IEEE Trans. Automat. Contr.*, vol. AC-24, pp.843-854, Dec. 1979.
- [11] M. Athans, R. H. Whiting, and M. Gruber, "A Suboptimal Estimation Algorithm with Probabilistic Editing for False Measurements with Application to Target Tracking with Wake Phenomena",

IEEE Trans. Automat. Contr., vol. AC-22, pp. 372-384, June 1977.

- [12] F. Burgeois, and J. C. Lasalle, "An Extension of the Munkres Algorithm for the Assignment Problem to Rectangular Matrices", *Communications of the ACM*, vol. 14, pp.802-806, Dec. 1971.
- [13] T. S. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association", *IEEE J. of Ocean Eng.*, vol. OC-9, pp.173-184, July 1983.
- [14] S. S. Lim, D. F. Liang, and M. Blanchette, "Air Defense Surveillance System Tracking Assessment", Proc. of the AG-ARD Conf. on Guidance and Control for Future Air-Defence Systems, Copenhagen, Denmark, May 17-20, 1994, AG-ARD-CP-555 (Suppliment:Classified papers), pp. 24.1-24.14, July 1995.
- [15] Y. Bar-Shalom(Ed.), *Multitarget-Multisensor Tracking: Applications and Advances*, vol. II, Artech House, 1992.
- [16] A. Kountzeris, *A Track Splitting Algorithm for Multiple Target Tracking*, University of Sussex, School of Eng. and Applied Sciences, Report CE/S/43, Oct. 1989.
- [17] P. Smith, and Buechler, "A Branching Algorithm for Discriminating and Tracking Multiple Objects", *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 101-104, Feb. 1975.
- [18] R. Danchick, and G. E. Newmam, "A Fast Method for Finding the Exact N-best Hypotheses for Multitarget Tracking", *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-29, pp.555-560, Apr. 1993.
- [19] M. Kovacich, T. Casaletto, B. Lutjens, D. McIntyre, R. Ralph, and E. VanDyk,

"An Application of MHT to Group to Object Tracking", SPIE vol. 1481, Signal and Data Processing of Small Targets 1991, pp.357-370, 1991.

기호설명

- k, k : 스캔변수
 k_f : 최종 스캔 번호 또는 최종 시각.
 N_{DT} : 구 표적에 배정된 측정치의 개수.
 N_{FT} : 허위 표적에 배정된 측정치의 개수.
 N_{TGT} : 이전에 확인된 표적의 총 개수로서 센서의 감시범위 내에 포함된 것.
 N_{NT} : 신 표적에 배정된 측정치의 개수.
 P_D : 표적검출 확률.
 β_{FT} : 허위표적의 밀도.
 β_{NT} : 신 표적의 밀도.
 $P_l^*(k)$: 시각 k 에서의 트래가설 l 의 확률.
 $L(k) \equiv (1+N_k) L(k-1)$: 스캔 k 에서의 총 트래가설의 수.
 $P_l(k, k-1)$: 가설 l 에 대한 추산오차의 분산.
 $K_l(k-1)$: 가설 l 에 대한 추산오차의 평균.
 $\hat{x}(k, k-1)$: 표적 상태 벡터의 추산치.
 $P(k, k)$: 추산오차의 분산.
 $A(k)$: 최적의 추적필터 이득.
 Y^k : 스캔 k 까지의 측정치의 총 집합.
 $Y(k)$: 스캔 k 에서의 측정치.
 $\Omega(\eta)$: 유효배정행렬
 ω : 사건 (event) 확률
 λ : 허위측정치의 밀도
 V : 판정영역
 ε : 연합사건(joint event)
 $S_j^*(k)$: \tilde{y}_j^* 의 covariance
 N_m : 측정치의 갯수
 N_l : 표적의 갯수
 $N_{la}(j)$: 허위표적을 제외한 표적으로서 측정치 j 에 배정된 것의 개수
 N_{l0} : 표적 l 에 배정되지 않은 측정치의 개수
 c : 정규화 상수
 τ_j : 측정치 j 가 어떤 표적에 배정되었는지를 나타내는 지표(1 또는 0).

δ_i : 표적 i 가 어떤 측정치에 배정되었는지를 나타내는 지표(1 또는 0).

임 상 석 (林尚石)



1955년 3월 15일 생

1976년 2월 한국항공대학교 항공전자공학과(공학사)

1984년 10월 Ottawa대학교 대학원(공학석사)

1990년 5월 Ottawa 대학교 대학원(공학박사)

1976년 3월~1982년 7월 국방과학연구소

1984년 11월~1987년 8월 캐나다 국방성 Defence Research Establishment Ottawa

1990년 6월~1992년 5월 Royal Military College

1992년 6월~1995년 2월 캐나다 국방성 DREO

1995년 3월~현재 한국항공대학교 항공전자공학과 조교수

관심분야 : CNS/ATM, 항행전자시스템, Target Tracking, Filter 설계 및 응용.