

적용 칼만 필터를 이용한 이동 표적 추적 기법

박 인환 조 경태
부산대학교 기계 공학과

Moving Target Tracking Technique
using adaptive Kalman Filter

I. H., Park K.R., Cho
Pusan National University

ABSTRACT

To track the manuevering target and to derive the Filter using state estimation and information in real time, we derive adaptive Kalman Filter which reinitialize the internal filter mode.

I. 서 론

MTI(Moving Target Indicator) 레이다 시스템에서 이동하는 목표물을 추적하고, 위치 정보를 얻기 위해서는 여러가지 동역학적 제어 기법이 사용되고 있으나, 알고리즘의 단순성과 안정성, 실시간 제어의 측면에서 Kalman 필터 기법이 대표적으로 사용된다.

기존의 Kalman 필터는 이동 목표물의 운동학적 상태 특성이 선형이면 주어진 정보로서 비교적 정확하게 목표물의 위치나 속도를 측정할수있다. 그러나 이동 목표의 상태 특성은 비선형 운동 방정식으로 해석되므로 Extended Kalman 필터의 응용이 시도되고 있으나, 비선형 특성때문에 필터의 분산, 발산등으로 안정성을 잃는 경우가 많으며, 전술비행(Maneuver) 하는 목표물의 지속적인 추적에 실패하는 경우가 많아 실제 MTI 레이다에서 적용하기는 많은 문제점을 야기하고 있다. 그러므로 전술비행하는 목표물의 위치 및 속도를 지속적으로 탐지하기 위하여, 추계학적인 관점에서 전술적 비행에 적용해가는 적응 필터의 개발이 요구된다.

전술적 비행을 하는 목표물의 추적이나 탐색법으로는 Jazwinski 의 Finite Memory 필터링법, Moose 의 n 개의 전략적 책략신호중 Semi-Markov 프로세서에 의한 입력신호 추정기법, 매개 변수들의 점프에 대한 적응 제어를 고려한 Sworder 의 필터링 기법, Mehra, Wilsky, Can 등 에 의한 Innovation 프로세서에 대한 Residual Sequence

테스트 기법등이 있으나, 현재 연구되고 있는 적응 필터의 개발 방향은 크게 두가지로 대별된다.

첫째는 추적 목표물의 가능한 Maneuver 를 Semi-Markov 프로세서를 가정하여, 그 프로세서의 가능한 조합으로 입력을 분석하여 Maneuver에 대처하는 것이고, 둘째는 측정된 목표물의 상태를 분석하여 maneuver를 감지한 후 보상 방법을 통하여 Maneuver에 대한 양 만큼 상태변수를 보상함으로써 빠른 시간내에 목표물의 상태를 추정하여, 지속적으로 필터를 구동하는 방법이다.

본 연구에서는 Maneuver가 없는 시스템의 상태 추정에 널리 이용되는 kalman 필터를 근본으로 하여, 이전 상태의 Innovation에 가중치를 가한 값의 합으로써 목표물의 Maneuver를 탐지한후, Kalman 필터의 상태를 바꿈으로써 Maneuver에 적용해가는 적응 Kalman 필터에 대하여 논하고, 현재 개발중인 MTI 레이다 시스템에 적용하고자 한다.

II. 본 론

1. 시스템의 구성

본 연구에서 제안한 추적 필터 시스템은 크게 보아 Maneuver가 존재하지 않는 주위 상황에서 구동되는 필터를 가진 동역학모델(Normal Model)과 Maneuver가 존재하는 주위상황에서 구동되는 동역학모델(Augmented Model), 그리고 Maneuver를 탐지하는 구조의 세가지로 구성된다.

1.1 Normal Model

일반적으로 Maneuver가 존재하지 않을때, 목표물은

동속도로 운동하며, 미세한 속도 변화를 외란 (Noise)으로 가지는 구조로 모델링되고 다음의 선형 추계학적인 이산 방정식으로 표시된다.

(1)

측정 방정식은

$$Z(K) = H X(K) + v(K) \quad (2)$$

으로 표시되며, $X(K) \in R^n$ 는 시스템 상태벡터로

$$X = [x \dot{x} y \dot{y}]^T \quad (3)$$

이며 초기치 $X(0)$ 는

$$\hat{X}(0) = E X(0) \quad (4)$$

$$P(0) = E X(0) X^T(0) \quad (5)$$

인 평균(X) 과 분산(P)를 갖는다.

$Z(K) \in R^r$ 는 측정 벡터이고, 시스템 입력 잡음 $W(K)$ 와 측정 잡음 $v(K)$ 는

$$E W(K) = 0 \quad E W(K) W^T(K) = Q(K) \quad (6)$$

$$E v(K) = 0 \quad E v(K) v^T(K) = R(K) \quad (7)$$

$$COV W(K) v(K) = E W(K) v^T(K) = 0 \quad (8)$$

을 갖는 White Gaussian 잡음으로 가정한다.

1.2 Augmented Model

일반적으로 Maneuver가 존재할때는 목표물은 등가속도로 운동하며, 가속도의 미세한 변화를 외란 (Noise)으로 가지는 구조로 모델링되며 다음과 같은 상태 방정식으로 표시된다.

$$X_m(K+1) = \Phi_m X_m(K) + G_m W_m(K) \quad (9)$$

$$Z_m(K) = H_m X_m(K) + v(K) \quad (10)$$

$X_m(K) \in R^n$ 는 시스템의 상태 벡터로

$$X_m = [x_m \dot{x}_m y_m \dot{y}_m \ddot{x}_m \ddot{y}_m] \quad (11)$$

이며 $W_m(K)$ 는

$$E W_m(K) = 0 \quad E W_m(K) W_m^T(K) = Q_m \quad (12)$$

을 갖는 White Gaussian 으로 가정한다.

1.3 Maneuver의 탐지

Normal Model에서 Maneuver를 탐지하려면

Maneuver 탐지 변수 $\mu(K)$ 를 가정한다.

$\mu(K)$ 는 Finite Data Window 상에서 Innovation 에 가중치를 곱한 값을 더하여서 결정한다.

$$\mu(K) = \alpha \mu(K-1) + \delta(K) \quad (13)$$

$$\delta(K) = V^T(K) S^{-1}(K) V(K) \quad (14)$$

여기서 $V(K)$ 는 Kalman Filter 의 Innovation 프로세서이며 $S(K)$ 는 Innovation의 분산이다.

또한 가중치는 $0 < \alpha < 1$ 이며 $\delta(K)$ 는 Gaussian Assumption 에 의해 n 자유도를 가지는 Chi-Square 분포로 가정한다.

$$\lim_{k \rightarrow \infty} E \mu(K) = n / (1 - \alpha) \quad (15)$$

그리고 Maneuver를 탐지하는 Finite Data Window 의 크기는

$$\Delta = (1 - \alpha)^{-1} \quad (16)$$

로 결정한다.

여기서 $\mu(K)$ 가 어떠한 Threshold N_m 보다 크면 Maneuver 가 일어난것이라 가정하고 추정 필터는 Augmented Model로 넘어간다.

Augmented Model 에서 Normal Model로 다시 전환 되기 위해서는, Finite data window 내에서의 가속도 분포가 Maneuver의 의미를 갖지 않아야 한다. 이것을 탐지하는 변수가 $\mu_a(K)$ 이며 다음과 같이 결정한다.

$$\delta a(K) = \hat{a}^T(K|K) P_m(K|K)^{-1} \hat{a}(K|K) \quad (17)$$

$$\mu_a(K) = \sum_{j=k-p+1}^k \delta a(j) \quad (18)$$

여기서 $P_m(K|K)$ 는 Error Covariance 행렬에서 가속도에 상응하는 분산 행렬이며, P 는 Finite Data Window의 크기이고 $\mu_a(K)$ 가 Threshold N_a 보다 적을때 가속도의 분포가 의미를 갖지 않으므로 Normal Model로 전환 된다. 그리고 여기서 가중치를 주지 않은 것은 가속도의 변화를 가정한 Augmented Model에서는 Maneuver에 대한 응답이 빠르기 때문이다.

2. Kalman 필터의 구성

시스템의 상태 벡터의 Maneuver가 일어나지 않을때, Kalman 필터는 이동하는 목표물을 추적하는데 널리 응용되고 있다.

Kalman 필터에서 상태 추정 Predictor 는

$$\hat{X}(K+1|K) = \Phi \hat{X}(K|K) \quad (19)$$

이고 Innovation 프로세서는

$$V(K) = Z(k) - H \hat{X}(K|K-1) \quad (20)$$

또서 White Gaussian 프로세서이다.
Corrector 는

$$\tilde{X}(K) = K(K) V(K) \quad (21)$$

이며 K(K)는 Kalman Gain 행렬이고, Estimator 는

$$\begin{aligned} \hat{X}(K|K) &= \hat{X}(K|K-1) + \tilde{X}(K) \\ &= \underline{\underline{\hat{X}}}(K-1|K-1) + K(K) V(K) \end{aligned} \quad (22)$$

이 된다.

여기서 " $\hat{\cdot}$ " 는 추정된 상태를 의미하고,
 $X(i|j)$ 는 $[X(i)|Z(j)]$ 를 약어를 쓴 표현이며
속정벡터 $Z(j)$ 를 조건부로 하여 $X(i)$ 추정함의
의미한다.

Predictor의 오차 분산은

$$P(K+1|K) = \underline{\underline{P}}(K|K) \underline{\underline{P}}^T + Q \quad (23)$$

Innovation $V(K)$ 의 오차 분산은

$$S(K) = H P(K|K-1) H^T + R \quad (24)$$

이며, Kalman 이득 $K(K)$ 는 다음 식으로
구해진다.

$$K(K) = P(K|K-1) H^T S^{-1}(K) \quad (25)$$

추정된 필터 $\hat{X}(K|K)$ 의 오차 분산은

$$P(K|K) = P(K|K-1) - K(K) H P(K|K-1) \quad (26)$$

이다.

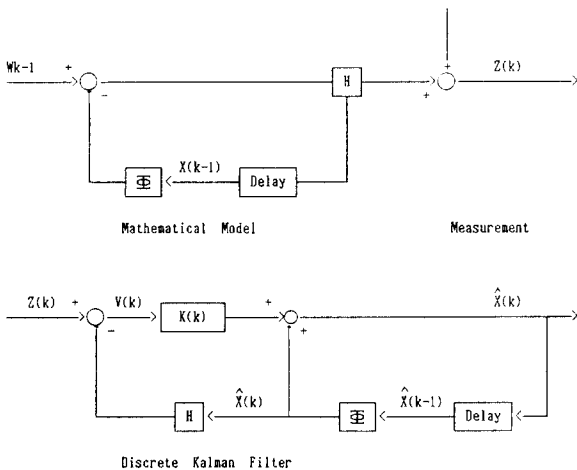


Fig. 1 System Model and Discrete Kalman Filter

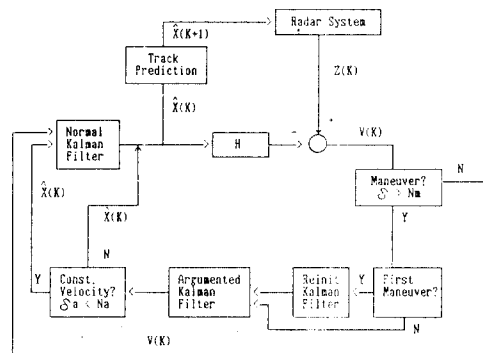
여기서 Kalman 필터 알고리즘을 다시 살펴보면
Kalman 필터가 Steady State 상태에 있을때
(23) - (26) 식의 오차 분산과 Kalman Gain
일정한 값으로 수렴하여 거의 불변하게 된다.
이때 Maneuver가 발생하면 (20) 식의 Innovation
은 큰 값을 가지게 되지만, (23) - (26) 식의
오차 분산을 증가 시켜서 Kalman Gain 을 증가
시켜줄 방법이 없어 상태 변수를 정확히 보상
하지 못해 Kalman 필터 알고리즘 기능을 제대
로 발휘하지 못한다.
그러므로, Maneuver가 일어나면 Kalman 필터
알고리즘 특성 때문에 실시간 추정 및 지속된
필터의 구동이 어려워진다.

3. 적응 필터의 구성

Maneuver가 발생 하였을때, 전술한 Maneuver
탐지 방법에 의해 Maneuver를 확인하고, System
의 모델을 Augmented Model로 바꾸어 Augmented
Kalman 필터를 구동한다.

그러면서 다시 가속도 변화를 확인한다.
가속도의 변화의 본포가 의미를 갖지 않을 때,
속 이동 표적이 등속도 운동을 하면 System
Model을 Normal Model로 바꾸어 Normal Kalman
필터를 구동하면서 이동 표적의 위치 정보를
계속 추정한다.

다음의 그림은 적응 Kalman 필터의 알고리즘을
설명한 것이다.



- $\hat{X}(K-1)$: Predicted State
- $\hat{X}(K)$: Estimated State
- $Z(K)$: Measurement
- H : Observation Matrix
- $V(K)$: Innovation
- δ : Maneuver Detection Factor
- δ_a : Constant Velocity Detection Factor
- δ_a : Maneuver Threshold
- δ_a : Constant Velocity Threshold

Fig. 2 Block Diagram of Adaptive Kalman Filter

III. 시뮬레이션

1. 시뮬레이션 모델

Normal 모델은

$$X(K+1) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X(K) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} W(K) \quad (27)$$

인 추계학적 다이내믹 시스템과

$$Z(K) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X(K) + v(K) \quad (28)$$

인 측정 방정식으로 주어지며, 샘플링 시간 T는 2 SEC 로 하고 Normal 모델에서 가중치 ω 는 0.75로 하여 데이터 Window의 크기는 4로 되게 하였다. 그리고, Threshold Nm과 Na를 Chi-Square 분포에서 95%의 신뢰도를 갖게 정해주면 각각 15.5, 9.49 이다.

시스템에 인가된 잡음은 X, Y 방향속도의 10%에 해당하는 White Gaussian Noise, 즉 분산이

$$E [W(k) W (k)] = Q(k) = (15.5 \text{ m/sec}) \quad (28)$$

$$\text{인 잡음을 인가하고 측정잡음은 속도를 감안해서} \quad E [(k) (k)] = R(k) = (12.5 \text{ m}) \quad (29)$$

로 주어지는 White Gaussian Noise 를 인가하였다.

2. 시뮬레이션 결과

목표물은 초기 상태

$$X(0) = [-900, 100 \text{ 2}, 0, 100 \text{ 2}] \quad (30)$$

에서 Fig. 3 같이 250 sec 동안 등속 비행 하였다.

또한 Simulation 결과는 Table 1 과 같다.

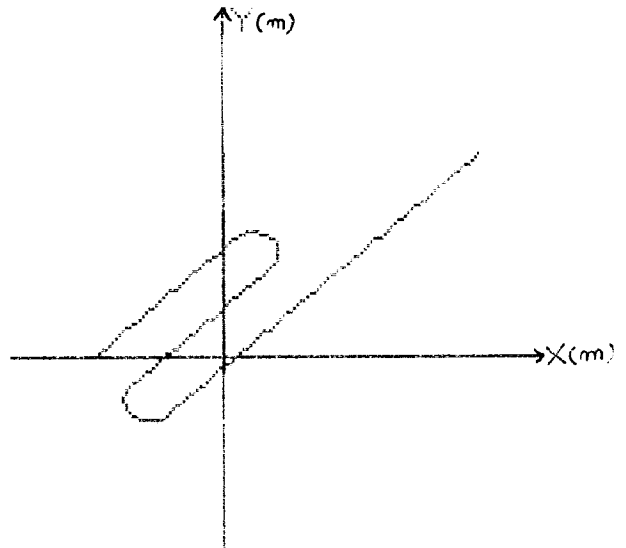


Fig. 3 Maneuvering Target Trajectory

Time (sec)	X (m)	Y (m)	Velocity (m/sec)	Acceleration (m/sec²)
0	-900	100	0	0
10	-880	100	10	0
20	-860	100	20	0
30	-840	100	30	0
40	-820	100	40	0
50	-800	100	50	0
60	-780	100	60	0
70	-760	100	70	0
80	-740	100	80	0
90	-720	100	90	0
100	-700	100	100	0
110	-680	100	110	0
120	-660	100	120	0
130	-640	100	130	0
140	-620	100	140	0
150	-600	100	150	0
160	-580	100	160	0
170	-560	100	170	0
180	-540	100	180	0
190	-520	100	190	0
200	-500	100	200	0
210	-480	100	210	0
220	-460	100	220	0
230	-440	100	230	0
240	-420	100	240	0
250	-400	100	250	0
260	-380	100	260	0
270	-360	100	270	0
280	-340	100	280	0
290	-320	100	290	0
300	-300	100	300	0
310	-280	100	310	0
320	-260	100	320	0
330	-240	100	330	0
340	-220	100	340	0
350	-200	100	350	0
360	-180	100	360	0
370	-160	100	370	0
380	-140	100	380	0
390	-120	100	390	0
400	-100	100	400	0
410	-80	100	410	0
420	-60	100	420	0
430	-40	100	430	0
440	-20	100	440	0
450	0	100	450	0
460	20	100	460	0
470	40	100	470	0
480	60	100	480	0
490	80	100	490	0
500	100	100	500	0
510	120	100	510	0
520	140	100	520	0
530	160	100	530	0
540	180	100	540	0
550	200	100	550	0
560	220	100	560	0
570	240	100	570	0
580	260	100	580	0
590	280	100	590	0
600	300	100	600	0
610	320	100	610	0
620	340	100	620	0
630	360	100	630	0
640	380	100	640	0
650	400	100	650	0
660	420	100	660	0
670	440	100	670	0
680	460	100	680	0
690	480	100	690	0
700	500	100	700	0
710	520	100	710	0
720	540	100	720	0
730	560	100	730	0
740	580	100	740	0
750	600	100	750	0
760	620	100	760	0
770	640	100	770	0
780	660	100	780	0
790	680	100	790	0
800	700	100	800	0
810	720	100	810	0
820	740	100	820	0
830	760	100	830	0
840	780	100	840	0
850	800	100	850	0
860	820	100	860	0
870	840	100	870	0
880	860	100	880	0
890	880	100	890	0
900	900	100	900	0

Table 1 Simulation Results of Maneuvering

Table 1 Continue

#	Prediction		Observation		Estimation		True Value		Target Speed #/sec	Range Error #	Bearing Error degree
	Range #	Bearing degree	Range #	Bearing degree	Range #	Bearing degree	Range #	Bearing degree			
44	10684	20.4	10580	20.4	10584	20.4	10584	20.1	-8.4	0.06	
45	10476	22.3	10332	21.9	10354	22.0	10354	20.8	-15.4	0.05	
46	10172	22.9	10052	22.6	10074	22.9	10074	206.0	-5.7	0.08	
47	9812	22.9	9721	22.7	9721	22.9	9721	198.0	-1.5	0.02	
48	9412	20.1	9338	20.1	9338	20.1	9338	197.9	-12.1	-0.09	
49	8885	20.1	8785	20.0	8785	20.0	8785	201.8	-17.1	-0.15	
50	8474	20.1	8352	20.1	8352	20.1	8352	202.8	-2.2	0.07	
51	8170	20.1	8083	20.1	8083	20.1	8083	203.1	5.8	0.08	
52	7841	20.1	7813	20.1	7813	20.1	7813	197.4	2.0	0.07	
53	7431	20.1	7439	20.1	7439	20.1	7439	197.4	13.6	0.10	
54	7071	20.1	7038	20.1	7038	20.1	7038	197.4	-1.7	0.08	
55	6972	20.1	6889	20.1	6889	20.1	6889	197.4	13.6	0.10	
56	6712	20.1	6637	20.1	6637	20.1	6637	197.4	-1.7	0.08	
57	6352	20.1	6269	20.1	6269	20.1	6269	197.4	13.6	0.10	
58	5992	20.1	5909	20.1	5909	20.1	5909	197.4	-1.7	0.08	
59	5632	20.1	5549	20.1	5549	20.1	5549	197.4	13.6	0.10	
60	5272	20.1	5189	20.1	5189	20.1	5189	197.4	-1.7	0.08	
61	4912	20.1	4829	20.1	4829	20.1	4829	197.4	13.6	0.10	
62	4552	20.1	4469	20.1	4469	20.1	4469	197.4	-1.7	0.08	
63	4192	20.1	4109	20.1	4109	20.1	4109	197.4	13.6	0.10	
64	3832	20.1	3749	20.1	3749	20.1	3749	197.4	-1.7	0.08	
65	3472	20.1	3389	20.1	3389	20.1	3389	197.4	13.6	0.10	
66	3112	20.1	3029	20.1	3029	20.1	3029	197.4	-1.7	0.08	
67	2752	20.1	2669	20.1	2669	20.1	2669	197.4	13.6	0.10	
68	2392	20.1	2309	20.1	2309	20.1	2309	197.4	-1.7	0.08	
69	2032	20.1	1949	20.1	1949	20.1	1949	197.4	13.6	0.10	
70	1672	20.1	1589	20.1	1589	20.1	1589	197.4	-1.7	0.08	
71	1312	20.1	1229	20.1	1229	20.1	1229	197.4	13.6	0.10	
72	952	20.1	869	20.1	869	20.1	869	197.4	-1.7	0.08	
73	592	20.1	509	20.1	509	20.1	509	197.4	13.6	0.10	
74	232	20.1	149	20.1	149	20.1	149	197.4	-1.7	0.08	
75	82	20.1	1	20.1	1	20.1	1	197.4	13.6	0.10	

TWS Target Data
MANUEVER #11

1-1-1980

TWS Target Data
MANUEVER #11

1-1-1980

IV. 결론

전술한 추적 시스템을 시뮬레이션한 결과 적응 필터의 상태는 대부분 Normal 모델 상태에서 작동하며, Maneuver가 탐지 되었을 때만 Augmented 모델로 전환되고, Maneuver를 탐지하기 위한 추적 계산량이 적으므로 실시간내에 목표물의 궤적을 탐지하고 추적 할 수 있었다. 결과에서 보듯이 Range error 는 0.5% 를 넘지 않으며 Bearing error 는 0.1 이내로서 목표물을 추적하는 System에 있어서는 충분하다고 생각 된다.

V. 참고문헌

- (1) ROBERT A SINGER, and KENNETH W.BEHNKE, "REAL-Time Tracking FILTER Evaluation and Selection for Trctical Applications"
- (2) CHAW-BING CHANG, and JKKOHN A.TABACZYWSKI " Application of State Estimation to Target Trackig" ,IEEE Trans. Automat. Contr., pp. 99-109, 1984
- (3) Y. BAR-SHALOM, and K.BIRMIWAL "VARIABLE DIMENSION Filter for Maneuvering Target Tracking,"IEEE Trans. Aerospace and Electronic Systems, vol AES-18, No.5,pp 621-629, SEPTEMBER 1982
- (4) Shame Series "Probability ", pp 188-216
- (5) Arthur Gelb, Applied Optimal Estimation MIT press