

비동기 다중레이더 환경에서 의사 측정치를 이용한 바이어스 추정기법

Multisensor Bias Estimation with Pseudo Measurement for Asynchronous Sensors

김형원* 김도형* 박효달** 송택렬*

Hyoung Won Kim Do Hyeung Kim Hyo Dal Park Taek Lyul Song

Abstract

In this paper, a sensor bias estimation method with pseudo measurement for asynchronous multisensor systems is proposed. The proposed bias estimation method separates the local filter which estimates the target state with biased measurements into two parts, one is bias part, the other is target state part. By using these two parts, the algorithm generates the pseudo bias measurement for estimating bias, and then eliminates bias of local track through bias compensation. Finally, the proposed algorithm is evaluated by comparing with the existing EXX method.

Keywords : Bias Estimation(바이어스 추정), Bias Compensation(바이어스 보상), Asynchronous Sensors(비동기 센서), Sensor Registration(센서 등록), Multi-target Tracking(다중표적 추적)

1. 서론

센서의 측정치 획득과정에서 생성되는 바이어스는 calibration error와 measurement bias와 같이 실제 표적의 측정치에 더해지거나 곱해진 상태로 발생할 수 있으며, moving platform에서 잘못된 센서의 위치 및 자세 정보를 이용할 경우나 또는 temporal bias와 같이 일시적으로 발생할 수 있다^[1]. 이러한 바이어스는 표

적 추적과정에서 추적 정밀도를 떨어뜨리는 원인이 되며 특히 다수의 센서를 이용한 표적 추적환경에서는 동일 표적에 대한 측정치를 서로 다른 표적으로 인식하거나 측정치 및 트랙을 이용한 융합^[2~6]시 융합성능을 저하시키는 요인이 된다. 이러한 이유로 다중센서를 이용한 표적 추적 시에는 바이어스 추정 및 보상이 필수적으로 요구되며 이러한 일련의 과정을 sensor registration이라고 부른다.

Sensor registration기법은 현재 많은 연구가 진행되고 있으며 이러한 방법 중 대표적으로 다중센서가 동기화된 상황에서 바이어스를 표적의 상태변수에 확장하여 추정하는 Augmented State Kalman Filter(ASKF)^[7]가 있으나 바이어스 항의 dimension증가에 따라 계산량이

† 2011년 9월 2일 접수~2011년 11월 25일 게재승인
 * 한양대학교(Hanyang University)
 ** 인하대학교(Inha University)
 책임저자 : 김형원(khwhy8858@naver.com)

증가하기 때문에 실제 적용이 어렵다는 단점을 지닌다. 이러한 단점을 보완하기 위하여 표적의 상태변수와 바이어스를 두 단계로 나누어 filtering하는 two-stage filter기법^[8]이 연구되었다. 최근에는 다수 센서에서 획득한 동일 표적의 측정치들을 이용하여 표적의 상태변수를 제거한 state independent pseudo measurement를 생성하고 이를 통해 바이어스를 추정하는 EX기법^[9]이 연구되었는데 최적화된 성능을 가진다고 알려져 있다. 하지만 이러한 기법들은 동기화 센서에서만 사용 가능한 방법으로 모든 센서가 동기화되기 어려운 실제 환경에 적용하기는 어렵다는 한계가 있다.

비동기 센서환경에서 바이어스를 추정하는 기법으로는 EX기법을 확장한 EXX기법^[10]이 존재하는데 이 방법은 서로 다른 시간대의 측정치들을 표적의 동역학 모델을 이용하여 state independent pseudo measurement를 생성하고 바이어스를 추정하는 기법으로 성능은 우수하나 바이어스를 쇄신하기 위해서는 proper time slot과 같은 기준으로 측정치를 모으는 과정이 필요하며 표적의 상태변수가 위치, 속도로 제한되어 있고 동역학 모델이 등속도 모델일 경우에만 사용 가능하다는 단점을 가진다.

본 논문에서 제안하는 바이어스 추정 알고리즘은 바이어스가 내포된 측정치를 이용하여 필터를 수행할 경우 필터의 추정치는 표적의 상태와 바이어스로 분리할 수 있다는 특징을 이용하여 의사 바이어스 측정치를 생성하고 바이어스를 추정한다. 또한 바이어스 보상 과정을 통하여 바이어스를 내포한 필터 추정치를 보정해 줌으로써 필터의 추정성능을 유지한다. 최종적으로 제안된 알고리즘은 기존의 EXX기법을 이용하여 그 성능을 비교 분석한다.

2. Sensor Bias Model

일반적인 센서의 측정치는 극좌표계에서 얻어지며 2차원의 환경에서 s 번째 센서의 표적 t 에 대한 측정치 모델은 다음과 같다.

$$z_{s,t(t_k)}^p = \begin{bmatrix} r_{s,t(t_k)}^p \\ \theta_{s,t(t_k)}^p \end{bmatrix} \quad (1)$$

$$= \begin{bmatrix} (1 + \epsilon_{s(t_k)}^r) r_{s,t(t_k)} + b_{s(t_k)}^r + w_{s(t_k)}^r \\ (1 + \epsilon_{s(t_k)}^\theta) \theta_{s,t(t_k)} + b_{s(t_k)}^\theta + w_{s(t_k)}^\theta \end{bmatrix}$$

여기서 $\epsilon_{s(t_k)}^r, \epsilon_{s(t_k)}^\theta$ 는 거리, 방위각에 대한 scale bias를 $b_{s(t_k)}^r, b_{s(t_k)}^\theta$ 는 거리, 방위각에 대한 offset bias를 의미하고, $w_{s(t_k)}^r, w_{s(t_k)}^\theta$ 는 거리, 방위각 성분의 측정잡음으로 평균이 0이고 분산이 $\sigma_{s(t_k)}^{r,2}, \sigma_{s(t_k)}^{\theta,2}$ 인 가우시안 분포를 한다. 표적 추적은 일반적으로 직교좌표계상에서 이루어지기 때문에 극좌표계의 측정치 식 (1)을 직교좌표계로 변환해야 하며 이는 식 (2)와 같이 근사화될 수 있다.

$$z_{s,t(t_k)}^m = HX_{t(t_k)} + B_{s,t(t_k)} C_{s,t(t_k)} \beta_{s(t_k)} + w_{s,t(t_k)}^m \quad (2)$$

여기서,

$$C_{s,t(t_k)} = \begin{bmatrix} 1 & 0 & \hat{r}_{s,t(t_k)} & 0 \\ 0 & 1 & 0 & \hat{\theta}_{s,t(t_k)} \end{bmatrix} \quad (3)$$

이며 $B_{s,t(t_k)}$ 는 jacobian matrix로

$$B_{s,t(t_k)} = \begin{bmatrix} \cos \hat{\theta}_{s,t(t_k)} & -\hat{r}_{s,t(t_k)} \sin \hat{\theta}_{s,t(t_k)} \\ \sin \hat{\theta}_{s,t(t_k)} & \hat{r}_{s,t(t_k)} \cos \hat{\theta}_{s,t(t_k)} \end{bmatrix} \quad (4)$$

와 같으며 식 (3), (4)에 들어가는 방위각과 거리는 측정치나 예측치를 사용하여 계산될 수 있다^[11]. 바이어스 벡터는 $\beta_{s(t_k)} = [b_{s(t_k)}^r, b_{s(t_k)}^\theta, \epsilon_{s(t_k)}^r, \epsilon_{s(t_k)}^\theta]^T$ 이며 표적에 대한 상태 벡터는 $X_{t(t_k)} = [x_{t(t_k)}, y_{t(t_k)}, \dot{x}_{t(t_k)}, \dot{y}_{t(t_k)}]^T$ 와 같이 위치와 속도 성분을 가진다. 측정치 행렬과 측정잡음의 오차공분산은 다음과 같다.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

$$R_{s,t(t_k)}^m = B_{s,t(t_k)} \begin{bmatrix} \sigma_{s(t_k)}^{r,2} & 0 \\ 0 & \sigma_{s(t_k)}^{\theta,2} \end{bmatrix} B_{s,t(t_k)}^T \quad (6)$$

3. Biased Local Filter

Biased local filter란 바이어스가 포함된 측정치를 이용한 표적추적 필터로 필터 추정치가 표적의 상태와

바이어스를 내포하고 있는 트랙을 의미한다. 먼저 필터의 동역학 모델은 다음과 같이 모델링된다.

$$X_{t(t_{k+1})} = \Phi_{(t_{k+1}, t_k)} X_{t(t_k)} + v_{(t_{k+1}, t_k)} \quad (7)$$

위 식에서 오른쪽 마지막항인 $v_{(t_{k+1}, t_k)}$ 는 공정잡음으로 평균이 0이고 공분산이 $Q_{(t_{k+1}, t_k)}$ 인 가우시안 분포를 가진다.

가. 트랙 초기화 과정

일반적으로 트랙 초기화는 연속된 시간의 2개 측정치를 이용하는 2 point differencing 기법이 많이 사용된다. 시간 t_0, t_1 에 s 번째 센서의 t 번째 표적 측정치를 이용한 로컬 트랙의 2 point differencing 기법은 다음과 같다.

$$\hat{X}_{s,t(t_1)}^b = \begin{bmatrix} z_{s,t(t_1)}^m \\ (z_{s,t(t_1)}^m - z_{s,t(t_0)}^m) / T \end{bmatrix}, T = t_1 - t_0 \quad (8)$$

여기서 $\hat{X}_{s,t(t_1)}^b$ 는 바이어스를 포함한 추정치를 의미하며 측정치모델 식 (2)를 이용하여 위 식을 정리하면 식 (9)와 같이 표적의 추정치와 바이어스 항으로 표현된다.(여기서 바이어스는 시간에 따라 변하지 않는다고 가정한다. $\beta_{s(t_1)} = \beta_{s(t_0)}$)

$$\begin{aligned} \hat{X}_{s,t(t_1|t_1)}^b &= \begin{bmatrix} HX_{t(t_1)} + w_{s,t(t_1)}^m \\ \{H(X_{t(t_1)} - X_{t(t_0)}) + (w_{s,t(t_1)}^m - w_{s,t(t_0)}^m)\} / T \\ B_{s,t(t_1)} C_{s,t(t_1)} \\ (B_{s,t(t_1)} C_{s,t(t_1)} - B_{s,t(t_0)} C_{s,t(t_0)}) / T \end{bmatrix} \beta_{s(t_1)} \\ &= \hat{X}_{s,t(t_1|t_1)} + G_{s,t(t_1)} \beta_{s(t_1)} \end{aligned} \quad (9)$$

$$G_{s,t(t_1)} = \begin{bmatrix} B_{s,t(t_1)} C_{s,t(t_1)} \\ (B_{s,t(t_1)} C_{s,t(t_1)} - B_{s,t(t_0)} C_{s,t(t_0)}) / T \end{bmatrix} \quad (10)$$

여기서 표적의 추정치인 $\hat{X}_{s,t(t_1|t_1)}$ 는 다음과 같은 오차 공분산을 가진다.

$$P_{s,t(t_1|t_1)} = \begin{bmatrix} R_{s,t(t_1)}^m & R_{s,t(t_1)}^m / T \\ R_{s,t(t_1)}^m / T & (R_{s,t(t_1)}^m + R_{s,t(t_0)}^m) / T^2 \end{bmatrix} \quad (11)$$

만약 bias free한 표적의 초기 추정치 및 오차공분산을 알 경우 식 (10)의 $G_{s,t(t_1)}$ 는 영행렬로 초기화 된다.

나. 트랙 예측 과정

t_k 시간의 바이어스를 포함하는 추정치가 다음과 같은 가우시안 분포를 할 경우

$$\hat{X}_{s,t(t_k|t_k)}^b \sim \mathcal{N}(\hat{X}_{s,t(t_k|t_k)}, P_{s,t(t_k|t_k)}) + G_{s,t(t_k)} \beta_{s(t_k)} \quad (12)$$

바이어스를 포함한 트랙의 예측과정은 필터의 동역학모델 식 (7)을 이용하여 다음과 같이 유도된다.

$$\begin{aligned} \hat{X}_{s,t(t_{k+1}|t_k)}^b &= \Phi_{(t_{k+1}, t_k)} \hat{X}_{s,t(t_k|t_k)}^b \\ &= \Phi_{(t_{k+1}, t_k)} \hat{X}_{s,t(t_k|t_k)} + \Phi_{(t_{k+1}, t_k)} G_{s,t(t_k)} \beta_{s(t_k)} \\ &= \hat{X}_{s,t(t_{k+1}|t_k)} + \Phi_{(t_{k+1}, t_k)} G_{s,t(t_k)} \beta_{s(t_k)} \end{aligned} \quad (13)$$

위 식에서처럼 바이어스를 포함하는 트랙의 예측치 역시 추정치와 바이어스 항으로 나누어 표현될 수 있으며 $\hat{X}_{s,t(t_{k+1}|t_k)}^b$ 의 오차공분산은 function of random variable를 이용하여 다음과 같이 구할 수 있다.

$$\begin{aligned} P_{s,t(t_{k+1}|t_k)}^b &= E \left[\begin{bmatrix} X_{s,t(t_{k+1})}^b - \hat{X}_{s,t(t_{k+1}|t_k)}^b \\ (X_{s,t(t_{k+1})}^b - \hat{X}_{s,t(t_{k+1}|t_k)}^b) T \end{bmatrix} \right] \\ &= E \left[\begin{bmatrix} \Phi_{(t_{k+1}, t_k)} \hat{e}_{s,t(t_k|t_k)} + v_{(t_{k+1}, t_k)} \\ (\Phi_{(t_{k+1}, t_k)} \hat{e}_{s,t(t_k|t_k)} + v_{(t_{k+1}, t_k)}) T \end{bmatrix} \right] \\ &= \Phi_{(t_{k+1}, t_k)} P_{s,t(t_k|t_k)} \Phi_{(t_{k+1}, t_k)}^T + Q_{(t_{k+1}, t_k)} \\ &= P_{s,t(t_{k+1}|t_k)} \end{aligned} \quad (14)$$

$$\hat{e}_{s,t(t_k|t_k)} = X_{s,t(t_k)} - \hat{X}_{s,t(t_k|t_k)} \quad (15)$$

식 (14)와 같이 예측치에 대한 오차공분산은 바이어스에 대한 영향을 받지 않는 형태로 표현된다.

다. 트랙 쇠신 과정

t_{k+1} 시간에 s 번째 센서의 표적 t 측정치인 $z_{s,t(t_{k+1})}^m$ 가 획득되었을 때 이에 해당하는 로컬 트랙의 쇠신 과정은 다음과 같다.

$$\begin{aligned} \hat{X}_{s,t(t_{k+1}|t_{k+1})}^b &= \hat{X}_{s,t(t_{k+1}|t_k)}^b + K_{s,t(t_{k+1})} \left(z_{s,t(t_{k+1})}^m - H\hat{X}_{s,t(t_{k+1}|t_k)}^b \right) \\ &= \hat{X}_{s,t(t_{k+1}|t_k)} + K_{s,t(t_{k+1})} \nu_{s,t(t_{k+1})} \\ &\quad + (I - K_{s,t(t_{k+1})}) H \Phi_{(t_{k+1}, t_k)} G_{s,t(t_k)} \beta_{s(t_k)} \\ &\quad + K_{s,t(t_{k+1})} B_{s,t(t_{k+1})} C_{s,t(t_{k+1})} \beta_{s(t_{k+1})} \\ &= \hat{X}_{s,t(t_{k+1}|t_{k+1})} + G_{s,t(t_{k+1})} \beta_{s(t_{k+1})} \end{aligned} \quad (16)$$

여기서,

$$\nu_{s,t(t_{k+1})} = HX_{t(t_{k+1})} + w_{s,t(t_{k+1})}^m - H\hat{X}_{s,t(t_{k+1}|t_k)} \quad (17)$$

$$K_{s,t(t_{k+1})} = P_{s,t(t_{k+1}|t_k)} H^T S_{s,t(t_{k+1})}^{-1} \quad (18)$$

$$S_{s,t(t_{k+1})} = H P_{s,t(t_{k+1}|t_k)} H^T + R_{s,t(t_{k+1})}^m \quad (19)$$

$$\begin{aligned} G_{s,t(t_{k+1})} &= (I - K_{s,t(t_{k+1})}) H \Phi_{(t_{k+1}, t_k)} G_{s,t(t_k)} \\ &\quad + K_{s,t(t_{k+1})} B_{s,t(t_{k+1})} C_{s,t(t_{k+1})} \end{aligned} \quad (20)$$

와 같다. 식 (16)과 같이 측정치를 이용하여 쇄신된 추정치 역시 표적의 추정치와 바이어스 항으로 나누어 표현되며 바이어스 행렬인 $G_{s,t(t_{k+1})}$ 는 식 (20)와 같이 재귀적으로 계산된다. 추정치의 오차공분산은 앞선 예측 오차공분산과 동일하게 계산되며 아래 수식과 같다.

$$\begin{aligned} P_{s,t(t_{k+1}|t_{k+1})}^b &= (I - K_{s,t(t_{k+1})}) P_{s,t(t_{k+1}|t_k)}^b \\ &= P_{s,t(t_{k+1}|t_{k+1})} \end{aligned} \quad (21)$$

바이어스가 실린 측정치를 이용한 추적필터는 초기화 과정부터 예측 및 쇄신과정까지 표적의 상태와 바이어스 항으로 분리될 수 있으며 바이어스 행렬인 $G_{s,t(t_k)}$ 또한 표적 추적과정에서 지속적으로 구할 수 있다.

4. Pseudo Bias Measurement

바이어스에 대한 의사 측정치는 바이어스 포함하는 센서 측정치와 로컬 트랙 추정치를 이용하여 생성할 수 있으며 이러한 의사 측정치는 특정 센서의 측정치

와 해당센서를 제외한 다른 센서의 로컬 트랙을 이용하여 생성한다. 먼저 t_k 시간에 표적 t 에 대한 i 번째 센서의 측정치 $z_{i,t(t_k)}^m$ 와 t_{k-1} 시간에 센서 j 의 로컬 트랙 추정치 $\hat{X}_{j,t(t_{k-1}|t_{k-1})}^b$ 를 이용한 의사 측정치는 다음과 같이 생성한다.

$$\begin{aligned} z_{t(t_k)}^{b(i,j)} &= z_{i,t(t_k)}^m - H\Phi_{(t_k, t_{k-1})} \hat{X}_{j,t(t_{k-1}|t_{k-1})}^b \\ &= B_{i,t(t_k)} C_{i,t(t_k)} \beta_{i(t_k)} - H\Phi_{(t_k, t_{k-1})} G_{j,t(t_{k-1})} \beta_{j(t_{k-1})} \\ &\quad + H\hat{e}_{j,t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \\ &= H_{t(t_k)}^{b(i,j)} \mathbf{b}_{t(t_k)}^{(i,j)} + w_{t(t_k)}^{b(i,j)} \end{aligned} \quad (22)$$

$$\hat{e}_{j,t(t_k|t_{k-1})} = X_{t(t_k)} - \hat{X}_{j,t(t_k|t_{k-1})} \quad (23)$$

여기서,

$$\mathbf{b}_{t(t_k)}^{(i,j)} = \begin{bmatrix} \beta_{i(t_k)} \\ \beta_{j(t_k)} \end{bmatrix} \quad (24)$$

$$H_{t(t_k)}^{b(i,j)} = [B_{i,t(t_k)} C_{i,t(t_k)} \quad -H\Phi_{(t_k, t_{k-1})} G_{j,t(t_{k-1})}] \quad (25)$$

$$w_{t(t_k)}^{b(i,j)} = H\hat{e}_{j,t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \quad (26)$$

의사 바이어스 측정치는 측정치 모델 식 (2)와 바이어스를 포함하는 예측치 식 (13)을 이용하여 얻을 수 있으며 식 (22)와 같이 바이어스 항과 새로운 측정잡음 항으로 정리 할 수 있다. 새로운 측정잡음의 오차공분산은 j 번째 로컬트랙의 예측치와 i 번째 센서의 측정잡음의 correlation이 없기 때문에 다음과 같이 유도된다.

$$\begin{aligned} \text{Cov} [w_{t(t_k)}^{b(i,j)}] &= R_{t(t_k)}^{b(i,j)} = E \left[\begin{pmatrix} H\hat{e}_{j,t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \\ \left(H\hat{e}_{j,t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \right)^T \end{pmatrix} \right] \\ &= H P_{j,t(t_k|t_{k-1})} H^T + R_{i,t(t_k)}^m \end{aligned} \quad (27)$$

만약 총 S 개의 센서가 존재할 경우 해당 표적에 대하여 S 개의 로컬 트랙의 존재하기 때문에 이에 대하여 의사 바이어스 측정치를 생성할 수 있다. 먼저 t_j 는 j 번째 센서의 로컬 트랙 쇄신시간을 의미하고 앞서 동일하게 t_k 시간에 표적 t 에 대한 i 번째 센서의 측

정치 $z_{i,t(t_k)}^m$ 를 입수하였을 경우 다음과 같이 stacked vector 형태로 의사 측정치를 생성한다.

$$z_{t(t_k)}^b = \begin{bmatrix} z_{t(t_k)}^{b(i,1)} \\ \vdots \\ (i \neq j) \\ \vdots \\ z_{t(t_k)}^{b(i,S)} \end{bmatrix} \quad (28)$$

여기서 i 번째 센서의 로컬 트랙을 제외한 $S-1$ 개의 의사 측정치를 생성할 수 있는데 위 식을 모든 센서의 stacked bias vector \mathbf{b} 로 다시 정리하면 다음과 같다.

$$z_{t(t_k)}^b = H_{t(t_k)}^b \mathbf{b} + w_{t(t_k)}^b \quad (29)$$

여기서,

$$\mathbf{b} = \begin{bmatrix} \beta_{1(t_k)} \\ \vdots \\ \beta_{S(t_k)} \end{bmatrix} \quad (30)$$

$$H_{t(t_k)}^b = \begin{bmatrix} -H\Phi_{(t,t)} G_{1,t(t_k)} \cdots B_{i,t(t_k)} C_{i,t(t_k)} \cdots & 0 \\ \vdots & \vdots \\ & (i \neq j) \\ \vdots & \vdots \\ 0 & \cdots B_{i,t(t_k)} C_{i,t(t_k)} \cdots -H\Phi_{(t,t_s)} G_{S,t(t_s)} \end{bmatrix} \quad (31)$$

$$w_{t(t_k)}^b = \begin{bmatrix} H\hat{e}_{1,t(t_k)} + w_{i,t(t_k)}^m \\ \vdots \\ (i \neq j) \\ \vdots \\ H\hat{e}_{S,t(t_k)} + w_{i,t(t_k)}^m \end{bmatrix} \quad (32)$$

와 같다. 의사 측정치의 측정 오차공분산은 식 (27)과 같이 추적필터의 오차공분산과 센서의 측정잡음의 합으로 표현되며 이는 필터의 추정성능에 따라 바이어스 추정성능이 결정될 수 있다는 것을 보여준다.

5. Bias Estimation

의사 바이어스 측정치를 이용한 바이어스 추정방법은 시간에 따라 바이어스가 변하지 않는 다는 가정하

에 EXX기법과 동일하게 Least Square Estimation(LSE)^[10]를 이용한다. 일반적으로 다중센서를 이용한 표적 추적은 단일 표적뿐만이 아니라 동시에 다수의 표적을 추적할 수 있기 때문에 이러한 환경을 고려해야 한다. 만약 총 N 개의 표적이 존재할 경우 각각의 센서는 N 개의 로컬 트랙을 가지게 되며 결국 센서의 센싱 주기마다 N 개의 의사 바이어스 측정치가 생성되게 된다. 이 중 t 번째 표적에 대한 바이어스 추정기법은 다음과 같다.

$$K_{t(t_k)}^b = \Sigma_{t-1(t_k)}^b H_{t(t_k)}^{bT} \left[H_{t(t_k)}^b \Sigma_{t-1(t_k)}^b H_{t(t_k)}^{bT} + R_{t(t_k)}^b \right]^{-1} \quad (33)$$

$$\hat{\mathbf{b}}_{t(t_k)} = \hat{\mathbf{b}}_{t-1(t_k)} + K_{t(t_k)}^b \left[z_{t(t_k)}^b - H_{t(t_k)}^b \hat{\mathbf{b}}_{t-1(t_k)} \right] \quad (34)$$

$$\Sigma_{t(t_k)}^b = \Sigma_{t-1(t_k)}^b - K_{t(t_k)}^b H_{t(t_k)}^b \Sigma_{t-1(t_k)}^b \quad (35)$$

여기서 $\hat{\mathbf{b}}_{t-1(t_k)}$, $\Sigma_{t-1(t_k)}^b$ 는 $t-1$ 번째 표적의 의사 바이어스 측정치를 이용하여 쇄신된 바이어스 추정치와 오차공분산을 의미한다. 최종적으로 t_k 시간에서의 바이어스 추정치는 $t=1 \sim N$ 까지 모든 표적에 대한 의사 바이어스 측정치를 이용하여 쇄신된 추정치가 된다.

$$\begin{aligned} \hat{\mathbf{b}}_{(t_k)} &= \hat{\mathbf{b}}_{N(t_k)} \\ \Sigma_{(t_k)} &= \Sigma_{N(t_k)} \end{aligned} \quad (36)$$

6. Bias Compensation

바이어스가 내포된 로컬 트랙의 추정치를 이용하여 비동기식 track to track융합^[6]을 하는 경우나 로컬 트랙의 추정 성능을 확인해야 하는 경우 필터의 추정치에 대한 바이어스 보상이 필수적으로 이루어져야 한다.

먼저 t_k 시간에 표적 t 에 대한 i 번째 센서의 로컬 트랙 추정치는 식 (16)와 같이 바이어스 항을 가지고 있으며 이 수식을 먼저 표적의 추정치 $\hat{X}_{i,t(t_k)t_k}$ 를 이용하여 정리하면 다음과 같다.

$$\hat{X}_{i,t(t_k)t_k} = \hat{X}_{i,t(t_k)t_k}^b - G_{i,t(t_k)} \beta_{i(t_k)} \quad (37)$$

위 식 (37)의 오른쪽 마지막에 있는 바이어스 $\beta_{i(t_k)}$ 는 실제 값을 알 수 없기 때문에 추정된 $\hat{\beta}_{i(t_k)}$ 를 이용하게 되며 이러한 바이어스 추정치는 평균이 $\hat{\beta}_{i(t_k)}$ 이고 오차공분산이 $\Sigma_{i(t_k)}^b$ 인 가우시안 분포를 가진다. 식 (37)에 바이어스 추정치를 대입하면 다음과 같이 정리된다.

$$\hat{X}_{i,t(t_k|t_k)}^{bias\ free} = \hat{X}_{i,t(t_k|t_k)}^{b} - G_{i,t(t_k)} \hat{\beta}_{i(t_k)} \quad (38)$$

바이어스가 제거된 추정치 $\hat{X}_{i,t(t_k|t_k)}^{bias\ free}$ 의 오차공분산은 다음과 같다.

$$\begin{aligned} P_{i,t(t_k|t_k)}^{bias\ free} &= E \left[\begin{pmatrix} X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)}^{bias\ free} \\ (X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)}^{bias\ free})^T \end{pmatrix} \right] \quad (39) \\ &= E \left[\begin{pmatrix} X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)}^b + G_{i(t_k)} \hat{\beta}_{i(t_k)} \\ (X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)}^b + G_{i(t_k)} \hat{\beta}_{i(t_k)})^T \end{pmatrix} \right] \\ &= E \left[\begin{pmatrix} X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)} + G_{i(t_k)} (\beta_{i(t_k)} - \hat{\beta}_{i(t_k)}) \\ (X_{i,t(t_k)} - \hat{X}_{i,t(t_k|t_k)} + G_{i(t_k)} (\beta_{i(t_k)} - \hat{\beta}_{i(t_k)}))^T \end{pmatrix} \right] \\ &= E \left[\begin{pmatrix} \hat{e}_{i,t(t_k|t_k)} + G_{i(t_k)} \hat{e}_{i,t(t_k)}^\beta \\ (\hat{e}_{i,t(t_k|t_k)} + G_{i(t_k)} \hat{e}_{i,t(t_k)}^\beta)^T \end{pmatrix} \right] \end{aligned}$$

$$\hat{e}_{i,t(t_k)}^\beta = \beta_{i(t_k)} - \hat{\beta}_{i(t_k)} \quad (40)$$

여기서 바이어스는 표적의 추정치와 correlation이 없다고 가정하여 다음과 같이 정리 할 수 있다.

$$P_{i,t(t_k|t_k)}^{bias\ free} = P_{i,t(t_k|t_k)} + G_{i,t(t_k)} \Sigma_{i(t_k)}^b G_{i,t(t_k)}^T \quad (41)$$

본 논문에서 제안하는 알고리즘의 전체적인 흐름도는 아래 그림과 같다.

Fig. 1과 같이 로컬 트랙을 이용한 바이어스 추정 및 보상 과정은 먼저 i 번째 센서에서 측정치가 입수되면 i 를 제외한 로컬트랙들의 추정치를 이용하여 의사 바이어스 측정치를 생성하고 바이어스를 추정한다. 센서 i 의 로컬트랙은 입수된 측정치를 이용하여 트랙을 쇄신하며 최종적으로 바이어스 보상과정을 통해 바이어스 영향을 최소화한 추정치를 얻게 된다.

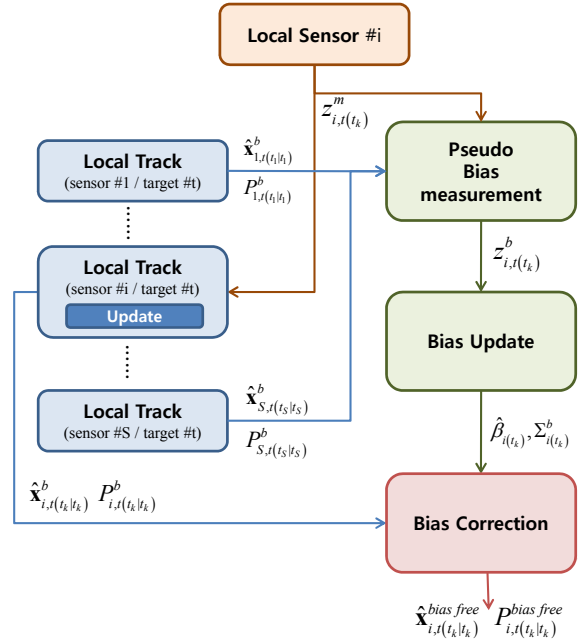


Fig. 1. Bias estimation and compensation method using biased local tracks

7. Simulation

시뮬레이션은 본 논문에서 제안한 바이어스 추정기법의 성능을 확인 위하여 비동기화 바이어스 추정기법 중 성능이 우수한 EXX기법을 대상으로 비교분석한다. 먼저 표적의 동역학 모델식 (7)의 상태천이 행렬은 아래와 같이 등속도 모델을 이용하며 공정잡음의 오차공분산은 Discretized Continuous White Noise Acceleration(DCWNA)모델을 이용하였다.

$$\Phi_{(t_{k+1}, t_k)} = \begin{bmatrix} I_2 & T I_2 \\ 0_2 & I_2 \end{bmatrix}, \quad T = t_{k+1} - t_k \quad (42)$$

$$Q_{(t_{k+1}, t_k)} = \begin{bmatrix} \frac{1}{3} T^3 I_2 & \frac{1}{2} T^2 I_2 \\ \frac{1}{2} T^2 I_2 & T I_2 \end{bmatrix} \tilde{q} \quad (43)$$

여기서 I_2 , 0_2 는 2×2의 단위행렬과 영행렬을 의미하며 \tilde{q} 는 power spectral density로 $6m^2/s^3$ 으로 두었다. 센서의 경우 두 개의 센서가 동일하게 1초의 주기를 가

하지만 0.5sec의 시간 offset을 가지는 비동기환경이다. 각각의 센서는 (0m, 0m)와 (10km, 0m)에 존재하며 두 센서의 측정잡음의 편차는 거리와 방위각 성분으로 $\sigma_1^r = \sigma_2^r = 10m$, $\sigma_1^\theta = \sigma_2^\theta = 1mrad$ 이며 각 센서의 바이어스는 거리, 방위각 방향의 offset bias와 거리, 방위각 방향의 scale bias 순으로 $\beta_1 = \beta_2 = [20m, 2mrad, 3 \times 10^{-5}m, 0.2mrad]^T$ 와 같다. 표적의 경우 총 32개의 표적이 x축과 y축 방향으로 각각 20m/s속도로 이동하며 표적의 시나리오는 다음 그림과 같다.

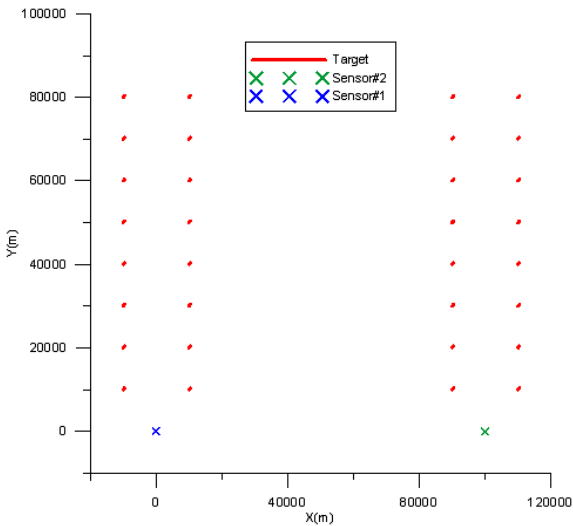


Fig. 2. Target Scenario

EXX기법과 제안한 알고리즘에서 바이어스 초기 추정치는 0 벡터로 두었으며 바이어스 초기 오차공분산은 $\Sigma_1^b = \Sigma_2^b = \text{diag}[(100)^2, (0.2)^2, (0.01)^2, (0.1)^2]$ 으로 초기화 하였다. 제안한 알고리즘에서 표적의 트랙은 bias free한 초기값을 알고 있다고 가정하며 실제 표적 상태에 평균이 0이고 분산이 초기 오차공분산인 가우시안 분포로 초기화한다. 여기서 초기 오차공분산은 모든 표적에 대해서 $P_{i,t(0|0)} = \text{diag}[10^2, 10^2, 5^2, 5^2]$ 로 두었으며 또한 바이어스 행렬인 $G_{i,t(0)}$ 는 0으로 초기화한다. 몬테칼로 시뮬레이션은 총 100회를 수행하였으며 바이어스를 추정된 결과는 다음과 같다.

다음 결과를 통해 볼 때 EXX기법 비하여 본 논문에서 제안한 방법이 보다 빠르게 바이어스를 추정하고 RMSE 오차도 향상시킨다는 것을 알 수 있다.

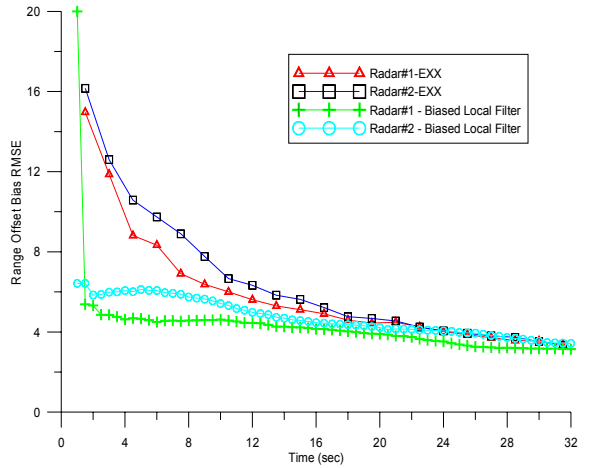


Fig. 3. Range offset bias RMSE

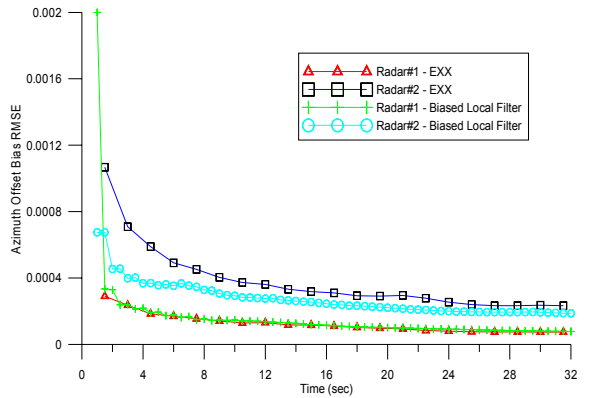


Fig. 4. Azimuth offset bias RMSE

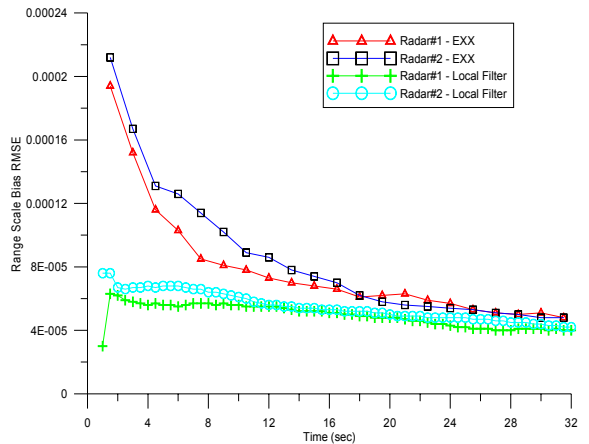


Fig. 5. Range scale bias RMSE

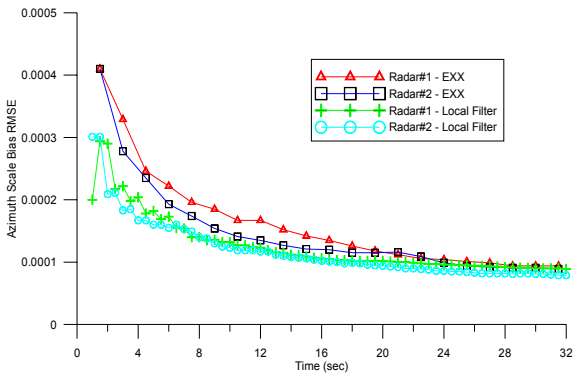


Fig. 6. Azimuth scale bias RMSE

다음은 바이어스 보상이 어느 정도 추정오차가 향상 되는지 확인해 보기 위하여 Sensor#1의 로컬 트랙 중 센서에서 거리가 가장 먼 Track#32과 가까운 Track#1의 추정오차 RMSE를 비교하였다. 다음 그림은 선택된 트랙을 나타낸 그림이다.

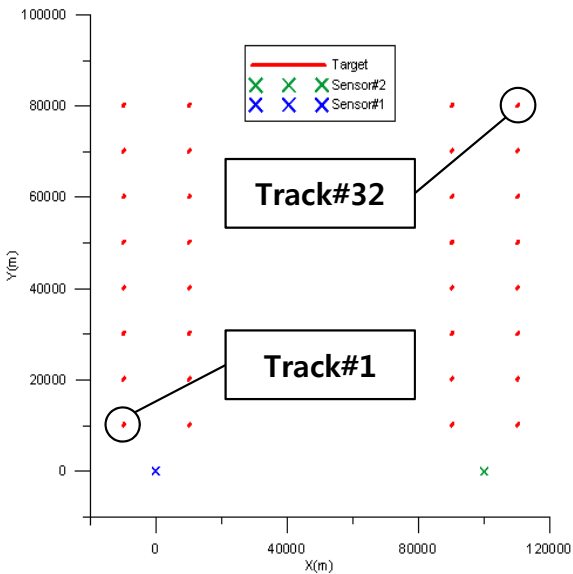


Fig. 7. Selected tracks

다음은 각 트랙의 RMSE의 결과이다.

다음 결과를 통하여 볼 때 바이어스 보상을 할 경우 바이어스 보상을 하지 않은 로컬 트랙에 비하여 위치와 속도 면에서 크게 향상되는 것을 볼 수 있다. 또한 바이어스로 인해 발생하는 오차의 크기는 거리와 각도에 비례하기 때문에 센서와 표적사이의 거리

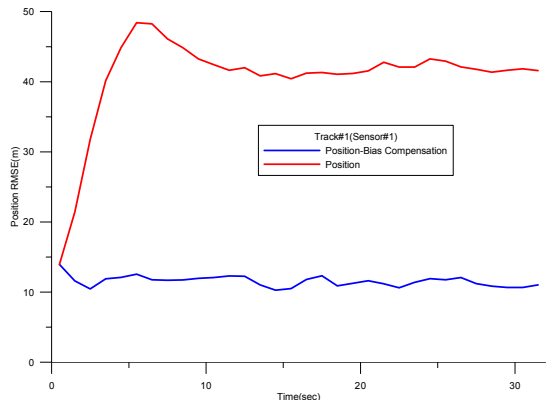


Fig. 8. Track#1(Sensor#1) position RMSE

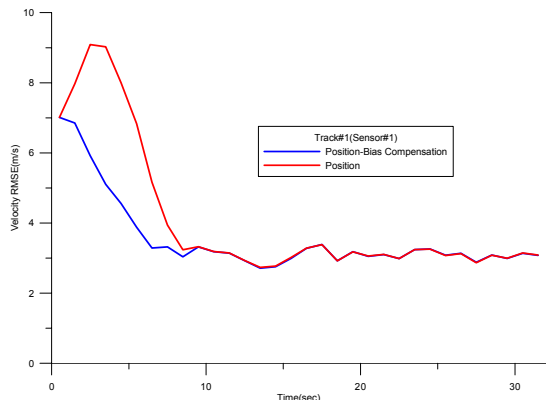


Fig. 9. Track#1(sensor#1) velocity RMSE

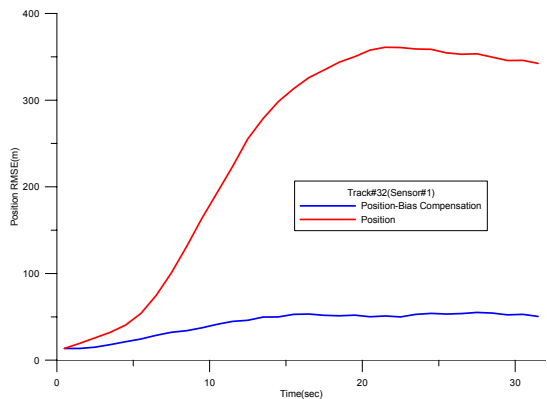


Fig. 10. Track#32(sensor#1) position RMSE

가 먼 Track#32가 Track#1 보다 위치 RMSE에 있어서 오차가 크게 나타나며 바이어스 보상에 있어서도 거리가 멀수록 보상되는 바이어스 값이 커지기 때문에 가

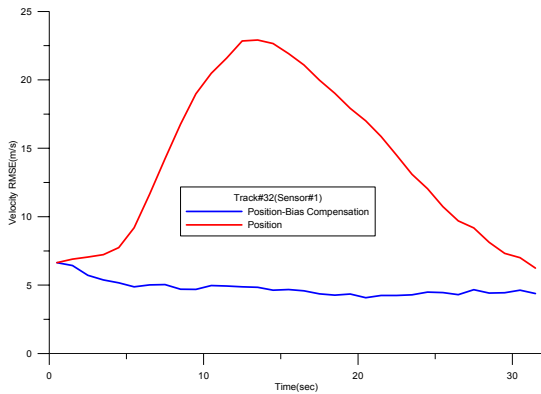


Fig. 11. Track#32(sensor#1) velocity RMSE

가운 표적을 추적하는 Track#1보다 Track#32가 향상된 바이어스 보정 성능을 보이게 된다.

8. 결론

본 논문에 제안한 바이어스 추정기법은 기존의 방법인 EXX기법보다 바이어스 추정성능이 우수하며 센서의 개수와 표적의 상태변수 및 동역학 모델에 따라 알고리즘이 달라지지 않는 장점을 지닌다. 또한 EXX기법은 proper time slot이라는 기준으로 측정치를 모아서 처리해야 하기 때문에 모든 센서에서 해당 표적에 대한 측정치가 입수되기 전까지 바이어스 추정을 하지 못하고 기다려야하는 단점을 지니지만 본 논문에서 제안하는 알고리즘은 측정치를 획득하게 되면 바로 바이어스를 쇄신 및 바이어스 보상이 이루어 질 수 있다는 장점을 가진다.

후 기

본 연구는 건설교통부 항공선진화사업의 연구비지원 (과제번호# 07항공-항행-03)에 의해 수행되었습니다.

References

[1] David F. Crouse, Yaakov Bar-Shalom, and Peter Willett, "Sensor Bias Estimation in the Presence of

Data Association Uncertainty", Signal and Data Processing of Small Targets, 2009. 8.
 [2] Qiang Gan and Chris J. Harris, "Comparison of Two Measurement Fusion Methods for Kalman-Filter-Based Multisensor Data Fusion", IEEE Transactions on Aerospace and Electronic Systems, Vol. 37, No. 1, pp. 273~280, 2001. 1.
 [3] K. C. Chang, Tian Zhi and R. K. Saha, "Performance Evaluation of Track Fusion with Information Matrix Filter", IEEE Transactions on Aerospace and Electronic Systems, Vol. 38, No. 2, pp. 455~466, 2002. 4.
 [4] K. C. Chang, R. K. Saha and Y. Bar-shalom, "On Optimal Track-to-Track Fusion", IEEE Transactions on Aerospace and Electronic Systems, Vol. 33, No. 4, pp. 1271~1276, 1997. 10.
 [5] Lucy Y. Pao and Christian W. Frei, "A Comparison of Parallel and Sequential Implementations of a Multisensor Multitarget Tracking Algorithm", America Control Conference Seattle, Washington, 1995. 6.
 [6] A. T. Alouani and T. R. Rice, "On Optimal Synchronous and Asynchronous Track Fusion", SPIE, Opt. Eng., Vol. 37, Issue 2, pp. 427~433, 1998. 2.
 [7] Friedland, B., "Treatment of Bias in Recursive Filtering", IEEE Transactions on Automatic Control, AC-140, 1969. 8.
 [8] A. T. Alouani, P. Xia, T. R. Rice, and W. D. Blair, "On the Optimality of Two-Stage State Estimation in the Presence of Random Bias", IEEE Trans. on Automatic Control, Vol. 38, No. 8, pp. 1279~1282, 1993. 8.
 [9] Xiangdong Lin, Bar-Shalom, Y, Kirubarajan, T., "Exact Multisensor Dynamic Bias Estimation with Local Tracks", IEEE Transactions on Aerospace and Electronic Systems, Vol. 40, No. 2, pp. 576~590, 2004. 4.
 [10] Xiangdong Lin, Bar-Shalom, Y, Kirubarajan, T., "Multisensor-Multitarget Bias Estimation for General Asynchronous Sensors", IEEE Transactions on Aerospace and Electronic Systems, Vol. 41, No. 3, pp. 899~921, 2005. 7.