순차적 정보융합을 이용한 비동기 다중 레이더 환경에서의 바이어스 추정기법

Multisensor Bias Estimation with Serial Fusion for Asynchronous Sensors

김 형 원* 박 효 달** 송 택 렬*
Hyoung Won Kim Hyo Dal Park Taek Lyul Song

Abstract

This paper presents a sensor bias estimation method with serial fusion for asynchronous multisensory systems. Serial fusion processes the sensor measurements in a first-come-first-serve basis and it plays an essential role in asynchronous fusion in practice. The proposed algorithm generates the bias measurements using fusion estimates and sensor measurements for bias estimation, and compensates the sensor biases in fusion tracks. A simulation study indicates that the proposed algorithm has the superior performance in bias estimation and accurate tracking.

Keywords : Sensor Registration(센서 등록), Bias Estimation(바이어스 추정), Asynchronous Sensors(비동기 센서), Serial Fusion(순차적 융합)

1. 서 론

일반적으로 다수의 센서를 이용하여 정보융합^[1-6]을 수행하기 위해서는 로컬 센서들이 갖는 다양한 좌표계를 하나의 공통된 tracking coordinate system으로 변환해주는 sensor registration이 필요하다. 여기서 각 로컬 센서가 가지는 바이어스는 registration error를 발생시키는데 이러한 registration error는 보정하지 않을 경우 트랙의 추정성능을 크게 하락시키며 동일표적에 대하여 다수의 트랙(ghosts)가 발생하는 등의 영향을 미치

게 되기 때문에 필수적으로 추정 및 보정이 이루어져야 한다.

바이어스를 추정하기 위한 기법으로는 Augmented State Kalman Filter(ASKF)와 같이 필터의 상태변수에 바이어스를 확장하는 고전적인 방법이 있으나 센서의 개수에 따라 상태변수가 증가되기 때문에 실제 구현이 어렵다는 단점을 가진다. [7]은 이러한 단점을 보완하기 위하여 바이어스와 그 이외에 상태변수를 분리하고 병렬로 two stage filter를 설계함으로써 필터의 차수를 줄였으며, [8]은 이러한 two stage filter를 일반화하였다.

[9]에서는 동기화 센서 환경에서 바이어스를 추정하기 위한 최적화된 방법으로 EX기법을 제시하였으나 비동기 환경이 주를 이루는 실제 환경에 적용하기에는 한계가 있다. 이러한 단점을 극복하기 위하여 [10]

† 2012년 5월 9일 접수~2012년 8월 24일 게재승인

* 한양대학교(Hanyang University)

** 인하대학교(Inha University)

책임저자 : 김형원(khwhy8858@naver.com)

에서는 비동기 환경에서 바이어스 추정을 하기위한 EXX기법을 제안하였다. EXX기법은 서로 다른 시간대의 측정치를 모아서 의사측정치를 생성하고 바이어스를 추정하는데 이때 측정치는 모든 센서에서 최소 하나 이상 입수되어야 한다는 proper time slot기준으로 모으게된다. 또한 EXX기법은 필터의 상태변수가 위치, 속도를 가지며 등속도 모델로 유도되었기 때문에 가속도와 같이 상태변수를 확장할 경우 알고리즘 적용이 불가능하다는 단점을 가진다.

[11]에서는 비동기 센서 환경에서 각각 로컬트랙을 이용하여 바이어스를 추정하고 추정한 바이어스를 이용하여 보정하는 방법에 대하여 소개하였다. [11]에서 제시된 방법은 EXX기법보다 바이어스 추정 성능이 우수하며 표적의 상태변수나 proper time slot과 같은 제한이 없다는 장점을 가진다. 하지만 하나의 바이어스 의사 측정치를 만들기 위해서 각 센서의 로컬트랙 추정치를 이용하기 때문에 많은 데이터 전송이 필요하며 융합된 추정결과를 얻기 위해서는 별도의 트랙 융합과정이 소요된다.

본 논문에서는 [11]과 다르게 순차적 정보융합기법을 통해 융합된 추정결과를 이용한 바이어스 추정 및 보정을 수행하는 기법에 대하여 소개한다. 먼저 2장에서는 센서의 바이어스 모델, 3장에서는 순차적 비동기 정보융합기법, 4장에서는 바이어스 의사 측정치 생성 과정을 5장에서는 바이어스 추정필터와 6장에서는 바이어스 보상기법에 대하여 소개하고 7장 시뮬레이션을 통하여 그 결과를 비교 분석하였다.

2. 센서 바이어스 모델

일반적으로 센서에서 얻어지는 측정치는 2차원 또는 3차원 환경에서 극좌표계로 획득되며 먼저 3차원 환경에서 s 번째 센서의 바이어스를 포함한 표적 t 의 측정치 모델은 다음과 같다.

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

여기서 s 는 센서의 인덱스, t 는 표적의 인덱스를 의미하며 t_k 는 해당 측정치가 입수된 시간을 의미한다. $\epsilon_{s(t_k)}^r, \epsilon_{s(t_k)}^\theta, \epsilon_{s(t_k)}^\psi$ 는 거리, 고각, 방위각에 대한 scale bias를 $b_{s(t_k)}^r, b_{s(t_k)}^\theta, b_{s(t_k)}^\psi$ 는 거리, 고각, 방위각에 대한 offset bias를 의미하고, $w_{s(t_k)}^r, w_{s(t_k)}^\theta, w_{s(t_k)}^\psi$ 는 거리, 고각, 방위각 성분의 측정잡음으로 평균이 0이고 분산이 $\sigma_{s(t_k)}^{r,2}, \sigma_{s(t_k)}^{\theta,2}, \sigma_{s(t_k)}^{\psi,2}$ 인 가우시안 분포를 한다. 표적 추적은 보통 직교좌표계상에서 이루어지기 때문에 극좌표계의 측정치 식 (1)을 직교좌표계로 변환해야 하며 이는 식 (2)와 같이 근사화^[12] 될 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,

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

이며 $B_{s,t(t_k)}$ 는 jacobian matrix로 (s, t, t_k notation 생략)

$$B = \begin{bmatrix} \cos \hat{\theta} \cos \hat{\psi} & -\hat{r} \sin \hat{\theta} \cos \hat{\psi} & -\hat{r} \cos \hat{\theta} \sin \hat{\psi} \\ \cos \hat{\theta} \sin \hat{\psi} & -\hat{r} \sin \hat{\theta} \sin \hat{\psi} & \hat{r} \cos \hat{\theta} \cos \hat{\psi} \\ \sin \hat{\theta} & \hat{r} \cos \hat{\theta} & 0 \end{bmatrix} \quad (4)$$

와 같으며 식 (3), (4)에 사용되는 거리, 고각 및 방위각은 측정치나 예측치를 이용할 수 있다.^[9] 바이어스는 $\beta_{s(t_k)} = [b_{s(t_k)}^r, b_{s(t_k)}^\theta, b_{s(t_k)}^\psi, \epsilon_{s(t_k)}^r, \epsilon_{s(t_k)}^\theta, \epsilon_{s(t_k)}^\psi]^T$ 이며 필터의 상태 변수는 $X_{t(t_k)} = [x_{t(t_k)}, y_{t(t_k)}, z_{t(t_k)}, \dot{x}_{t(t_k)}, \dot{y}_{t(t_k)}, \dot{z}_{t(t_k)}]^T$ 와 같이 위치, 속도성분을 가진다. 식 (2)의 측정치 행렬과 측정잡음의 오차공분산은 다음과 같다.

$$H = [I_3 \ 0_3] \quad (5)$$

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

여기서 $I_3, 0_3$ 은 3x3의 단위행렬과 영행렬을 의미한다.

3. 순차적 비동기 정보융합 방법

비동기 센서 환경에서 이루어지는 정보융합기법은 동기화 환경에서 이루어지는 정보융합기법^[6]과 동일하게 중앙집중식과 분산식으로 분리 할 수 있으며 분산식 방법에서는 각 로컬 센서로부터 추적되는 트랙들을 융합하는 asynchronous track to track fusion기법^[5] 등이 존재한다. 본 논문에서는 이러한 다양한 융합방식 중 로컬 센서로부터 입수되는 측정치를 중앙에서 순차적으로 처리하는 순차적 비동기 정보융합기법^[4]을 이용하여 바이어스 추정 및 보정기법을 설계하였으며 먼저 융합 알고리즘 흐름도는 다음과 같다.

Fig. 1. Asynchronous serial fusion flowchart

위 흐름도처럼 먼저 t_{k-2} 시간에 센서2의 표적 t 에 대한 측정치 $z_{2,t(t_{k-2})}^m$ 와 측정잡음 $R_{2,t(t_{k-2})}^m$ 가 입수되면 융합센서의 트랙에서는 해당 데이터를 이용하여 상태변수를 채신하게 되는데 이는 standard Kalman filter의 채신식과 동일하다. 이후 t_{k-1} 시간에 센서1에서 표적에 대한 측정치를 입수하게 되면 융합트랙은 t_{k-1} 시간까지 예측을 수행하고 센서1의 데이터를 이용하여 상태변수를 채신하며 이와 같은 방법으로 각 센서에서 들어오는 측정 데이터를 융합센서에 받아서 순서대로 처리하는 방식이다. 여기서 동일시간에 다수의 센서에서 측정치가 들어오는 경우 처음은 예측을 수행하여 상태변수를 채신하지만 그 이후부터는 예측을 생략하고 채신된 추정치를 예측치로 두어 상태변수 채신을 수행한다.^[5]

융합필터의 예측에 사용되는 표적의 동역학 모델은 다음과 같이 선형모델을 사용한다.

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

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. 융합트랙 초기화 과정

일반적으로 트랙 초기화는 연속된 시간에 동일 표적의 측정치 2개가 존재할 때 각 측정치의 위치 및 위치차이를 이용하여 필터의 상태변수를 초기화하는 2 point differencing기법이 사용된다. Fig. 1과 같이 순차적 비동기 정보융합에서 트랙초기화를 하는 경우 동일 센서가 아닌 다른 센서의 측정치들을 이용할 수 있는데 이때 서로 다른 센서의 바이어스로 인하여 위치 차이가 커지며 이는 초기 속도추정치에 크게 영향을 미치게 된다. 하지만 동일 센서의 측정치를 이용하여 초기화하는 경우 해당 측정치들에 포함되는 바이어스는 거의 유사하며 두 측정치의 위치 차이를 구할 때 바이어스가 서로 상쇄되기 때문에 바이어스에 대한 영향을 최소화할 수 있으므로 본 논문에서는 동일 센서의 측정치를 이용한 트랙초기화 방법을 이용한다. s 번째 센서의 측정치가 입수되었을 경우 이를 이용한 융합트랙의 상태변수는 다음과 같이 초기화된다. 여기서 윗첨자 b 는 상태변수에 바이어스를 포함되어 있다는 것을 의미한다.

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

본 논문에서 바이어스 항은 시간에 따라 변하지 않는다고 가정하며 총 S 개의 로컬센서가 존재할 때 바이어스 벡터 $\mathbf{b}$ 는 다음과 같이 stacked vector 형태로 정의할 경우

$$\mathbf{b} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_s \\ \vdots \\ \beta_S \end{bmatrix} \quad (9)$$

식 (8)에서 바이어스를 포함한 융합추정치 $\hat{X}_{t(t_i)}^b$ 는 측정치모델 식 (2)와 식 (9)를 이용하여 정리하면 다음과 같이 표적의 추정치와 바이어스 벡터항으로 분리될 수 있다.

$$\begin{aligned} \hat{X}_{t(t_1|t_1)}^b &= \left[\begin{array}{c} HX_{t(t_1)} + w_{s,t(t_1)}^m \\ \{H(X_{t(t_1)} - X_{t(t_0)}) + (w_{s,t(t_1)}^m - w_{s,t(t_0)}^m)\} / T \\ + [0_6 \cdots G_{s,t(t_1)} \cdots 0_6] \mathbf{b} \end{array} \right] \\ &= \hat{X}_{t(t_1|t_1)} + G_{t(t_1)}^f \mathbf{b} \end{aligned} \quad (10)$$

$$G_{s,t(t_1)} = \left[\begin{array}{c} B_{s,t(t_1)} C_{s,t(t_1)} \\ (B_{s,t(t_1)} C_{s,t(t_1)} - B_{s,t(t_0)} C_{s,t(t_0)}) / T \end{array} \right] \quad (11)$$

여기서 0_6 은 6×6 의 영행렬을 의미하며 바이어스를 알고 있는 상수라고 가정할 경우 초기 융합추정치 $\hat{X}_{t(t_1|t_1)}^b$ 는 다음과 같은 오차 공분산을 가진다.

$$P_{t(t_1|t_1)}^b = \left[\begin{array}{cc} R_{s,t(t_1)}^m & R_{s,t(t_1)}^m / T \\ R_{s,t(t_1)}^m / T & (R_{s,t(t_1)}^m + R_{s,t(t_0)}^m) / T^2 \end{array} \right] \quad (12)$$

만약 bias free한 표적의 초기 추정치 및 오차공분산을 알 경우에는 식 (11)의 $G_{s,t(t_1)}$ 는 영행렬로 초기화된다.

나. 융합트랙 예측 과정

t_k 시간의 바이어스를 포함하는 융합 추정치가 다음과 같은 가우시안 분포를 할 경우

$$\hat{X}_{t(t_k|t_k)}^b \sim \mathcal{N}(\hat{X}_{t(t_k|t_k)}, P_{s,t(t_k|t_k)}) + G_{t(t_k)}^f \mathbf{b} \quad (13)$$

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

$$\begin{aligned} \hat{X}_{t(t_{k+1}|t_k)}^b &= \Phi_{(t_{k+1}, t_k)} \hat{X}_{t(t_k|t_k)}^b \\ &= \Phi_{(t_{k+1}, t_k)} \hat{X}_{t(t_k|t_k)} + \Phi_{(t_{k+1}, t_k)} G_{t(t_k)}^f \mathbf{b} \\ &= \hat{X}_{t(t_{k+1}|t_k)} + \Phi_{(t_{k+1}, t_k)} G_{t(t_k)}^f \mathbf{b} \end{aligned} \quad (14)$$

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

$$\begin{aligned} P_{t(t_{k+1}|t_k)}^b &= \left[\begin{array}{c} (X_{t(t_k)}^b - \hat{X}_{t(t_k)}^b) \\ (X_{t(t_k)}^b - \hat{X}_{t(t_k)}^b)^T \end{array} \right] \\ &= \left[\begin{array}{c} (\Phi_{(t_{k+1}, t_k)} \hat{e}_{t(t_k|t_k)} + v_{(t_{k+1}, t_k)}) \\ (\Phi_{(t_{k+1}, t_k)} \hat{e}_{t(t_k|t_k)} + v_{(t_{k+1}, t_k)})^T \end{array} \right] \\ &= \Phi_{(t_{k+1}, t_k)} P_{t(t_k|t_k)} \Phi_{(t_{k+1}, t_k)}^T + Q_{(t_{k+1}, t_k)} \\ &= P_{t(t_{k+1}|t_k)} \end{aligned} \quad (15)$$

$$\hat{e}_{t(t_k|t_k)} = X_{t(t_k)} - \hat{X}_{t(t_k|t_k)} \quad (16)$$

다. 융합트랙 쇄신 과정

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

$$\begin{aligned} \hat{X}_{t(t_{k+1}|t_{k+1})}^b &= \hat{X}_{t(t_{k+1}|t_k)}^b + K_{t(t_{k+1})} (z_{s,t(t_{k+1})}^m - H \hat{X}_{t(t_{k+1}|t_k)}^b) \\ &= \hat{X}_{t(t_{k+1}|t_k)} + K_{t(t_{k+1})} \nu_{s,t(t_{k+1})} \\ &\quad + (I - K_{t(t_{k+1})} H) \Phi_{(t_{k+1}, t_k)} G_{t(t_k)}^f \mathbf{b} \\ &\quad + K_{t(t_{k+1})} B_{s,t(t_{k+1})} C_{s,t(t_{k+1})} \beta_s \end{aligned} \quad (17)$$

여기서 측정잔차, 칼만이득 및 측정잔차의 공분산은 아래와 같으며

$$\nu_{s,t(t_{k+1})} = HX_{t(t_{k+1})} + w_{s,t(t_{k+1})}^m - H \hat{X}_{t(t_{k+1}|t_k)}^b \quad (18)$$

$$K_{t(t_{k+1})} = P_{t(t_{k+1}|t_k)} H^T S_{t(t_{k+1})}^{-1} \quad (19)$$

$$S_{t(t_{k+1})} = H P_{t(t_{k+1}|t_k)} H^T + R_{s,t(t_{k+1})}^m \quad (20)$$

식 (17) 등호의 오른쪽 마지막 항을 $\mathbf{b}$ 로 정리하면 다음과 같다.

$$K_{t(t_{k+1})} B_{s,t(t_{k+1})} C_{s,t(t_{k+1})} \beta_s = K_{t(t_{k+1})} B C_{s,t(t_{k+1})}^f \mathbf{b} \quad (21)$$

$$B C_{s,t(t_{k+1})}^f = [0_{3,6} \cdots B_{s,t(t_{k+1})} C_{s,t(t_{k+1})} \cdots 0_{3,6}] \quad (22)$$

여기서 $0_{3,6}$ 은 3×6 의 영행렬을 의미한다. 바이어스가 포함된 융합추정치 식 (17)에 식 (21)을 대입하여 다시 정리하면

$$\begin{aligned}\hat{X}_{t(t_{k+1}|t_{k+1})}^b &= \hat{X}_{t(t_{k+1}|t_k)} + K_{t(t_{k+1})} \nu_{s,t(t_{k+1})} \\ &\quad + (I - K_{t(t_{k+1})} H) \Phi_{(t_{k+1}, t_k)} G_{t(t_k)}^f \mathbf{b} \\ &\quad + K_{t(t_{k+1})} B C_{s,t(t_{k+1})}^f \mathbf{b} \\ &= \hat{X}_{t(t_{k+1}|t_{k+1})} + G_{t(t_{k+1})}^f \mathbf{b}\end{aligned}\quad (23)$$

$$\begin{aligned}G_{t(t_{k+1})}^f &= (I - K_{t(t_{k+1})} H) \Phi_{(t_{k+1}, t_k)} G_{t(t_k)}^f \\ &\quad + K_{t(t_{k+1})} B C_{s,t(t_{k+1})}^f\end{aligned}\quad (24)$$

와 같다. 식 (23)과 같이 추정치를 이용하여 쇄신된 추정치 역시 표적의 추정치와 바이어스 항으로 나누어 표현되며 바이어스 행렬인 $G_{t(t_{k+1})}^f$ 는 식 (24)와 같이 재귀적으로 계산된다. 추정치의 오차공분산은 앞선 예측 오차공분산과 동일하게 계산되며 아래 수식과 같다.

$$\begin{aligned}P_{t(t_{k+1}|t_{k+1})}^b &= (I - K_{s,t(t_{k+1})} H) P_{t(t_{k+1}|t_k)}^b \\ &= P_{t(t_{k+1}|t_{k+1})}\end{aligned}\quad (25)$$

바이어스가 실린 추정치를 이용한 추적필터는 초기화 과정부터 예측 및 쇄신과정까지 표적의 상태와 바이어스 항으로 분리될 수 있으며 바이어스 행렬인 $G_{t(t_{k+1})}^f$ 또한 표적 추적과정에서 지속적으로 구할 수 있다.

4. 바이어스 의사 추정치

본 논문에서는 바이어스를 추정하기 위해 바이어스에 대한 의사 추정치를 생성하는데 이는 융합트랙의 추정치가 표적의 상태항과 바이어스 항으로 나누어지는 점을 이용하여 구하게 된다. 먼저 t_{k-1} 시간에 표적 t 에 대한 융합트랙 추정치 $\hat{X}_{t(t_{k-1}|t_{k-1})}^b$ 와 t_k 시간에 i 번째 센서의 추정치 $z_{i,t(t_k)}^m$ 를 이용한 의사 추정치는 다음과 같이 생성한다.

$$\begin{aligned}z_{i,t(t_k)}^b &= z_{i,t(t_k)}^m - H \Phi_{(t_k, t_{k-1})} \hat{X}_{t(t_{k-1}|t_{k-1})}^b \\ &= B C_{i,t(t_k)} \mathbf{b} - H \Phi_{(t_k, t_{k-1})} G_{t(t_{k-1})}^f \mathbf{b} \\ &\quad + H \hat{e}_{t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \\ &= H_{i,t(t_k)}^b \mathbf{b} + w_{i,t(t_k)}^b\end{aligned}\quad (26)$$

$$\hat{e}_{t(t_k|t_{k-1})} = X_{t(t_k)} - \hat{X}_{t(t_k|t_{k-1})}\quad (27)$$

여기서 바이어스에 대한 새로운 추정치 행렬과 추정잡음은 다음과 같다.

$$H_{i,t(t_k)}^b = B C_{i,t(t_k)} - H \Phi_{(t_k, t_{k-1})} G_{t(t_{k-1})}^f\quad (28)$$

$$w_{i,t(t_k)}^b = H \hat{e}_{t(t_k|t_{k-1})} + w_{i,t(t_k)}^m\quad (29)$$

식 (29)의 바이어스 의사 추정치의 추정잡음 오차공분산은 다음과 같이 유도된다.

$$\begin{aligned}\text{Cov}[w_{i,t(t_k)}^b] &= R_{i,t(t_k)}^b = E \left[\begin{pmatrix} H \hat{e}_{t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \\ H \hat{e}_{t(t_k|t_{k-1})} + w_{i,t(t_k)}^m \end{pmatrix}^T \right] \\ &= H P_{t(t_k|t_{k-1})} H^T + R_{i,t(t_k)}^m\end{aligned}\quad (30)$$

위 추정 오차공분산은 추정잔차의 오차공분산 식 (20)과 동일하다는 것을 알 수 있으며 이는 융합트랙 쇄신과정에서 쉽게 계산될 수 있다는 장점을 가지게 된다. 또한 위 식 (30)에서 오차공분산은 추적필터의 오차공분산과 센서의 추정잡음의 합으로 표현되기 때문에 필터의 추정성능에 따라 바이어스 추정성능이 결정될 수 있다는 것을 보여준다.

5. 바이어스 추정 필터

바이어스 추정방법은 EXX기법과 동일하게 Least Square Estimation(LSE)^[10]를 사용할 수 있다. 먼저 바이어스는 시간에 따라 변하지 않는다는 가정하에 이전 시간의 바이어스 추정치는 예측치로 치환되어 사용된다. 또한 다수 표적 관점에서 총 N 개의 표적이 존재할 때 N 개의 융합트랙이 존재하게 되며 바이어스 의사 추정치도 각 센서의 주기마다 총 N 개가 생성되기 때문에 이를 이용하여 바이어스 추정이 이루어져야 한다. 먼저 총 N 개의 표적 중 t 번째 표적에 대한 의사 추정치를 이용하여 바이어스 추정은 다음과 같다.

$$K_{t(t_k)}^b = \Sigma_{t-1(t_k)}^b H_{t(t_k)}^{b^T} \left[H_{t(t_k)}^b \Sigma_{t-1(t_k)}^b H_{t(t_k)}^{b^T} + R_{t(t_k)}^b \right]^{-1}\quad (31)$$

$$\hat{\mathbf{b}}_{t(t_k)} = \hat{\mathbf{b}}_{t-1(t_k)} + K_{t(t_k)}^b \left[z_{t(t_k)}^b - H_{t(t_k)}^b \hat{\mathbf{b}}_{t-1(t_k)} \right] \quad (32)$$

$$\Sigma_{t(t_k)}^b = \Sigma_{t-1(t_k)}^b - K_{t(t_k)}^b H_{t(t_k)}^b \Sigma_{t-1(t_k)}^b \quad (33)$$

여기서 $\hat{\mathbf{b}}_{t-1(t_k)}$, $\Sigma_{t-1(t_k)}^b$ 는 $t-1$ 번째 표적의 의사 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.

$$\hat{\mathbf{b}}_{(t_k)} = \hat{\mathbf{b}}_{N(t_k)}, \Sigma_{(t_k)}^b = \Sigma_{N(t_k)}^b \quad (34)$$

6. 바이어스 보상

바이어스를 내포하고 있는 융합트랙의 바이어스 보상 방법은 다음과 같다. 먼저 t_k 시간에 표적 t 에 대한 융합추정치는 식 (23)과 같이 바이어스 항을 가지고 있으며 이 수식을 먼저 표적의 추정치 $\hat{X}_{i,t(t_k|t_k)}$ 를 이용하여 정리하면 다음과 같다.

$$\hat{X}_{t(t_k|t_k)} = \hat{X}_{t(t_k|t_k)}^b - G_{t(t_k)}^f \mathbf{b} \quad (35)$$

위식에서 오른쪽 마지막에 있는 바이어스 벡터 $\mathbf{b}$ 는 실제 값을 알 수 없기 때문에 추정된 $\hat{\mathbf{b}}_{(t_k)}$ 를 이용하게 되며 이러한 바이어스 추정치는 평균이 $\hat{\mathbf{b}}_{(t_k)}$ 이고 오차 공분산이 $\Sigma_{i(t_k)}^b$ 인 가우시안 분포를 가진다. 식 (35)에 이러한 바이어스 추정치를 대입하면

$$\hat{X}_{t(t_k|t_k)}^{bias\ free} = \hat{X}_{t(t_k|t_k)}^b - G_{t(t_k)}^f \hat{\mathbf{b}}_{(t_k)} \quad (36)$$

바이어스를 보상한 추정치 $\hat{X}_{t(t_k|t_k)}^{bias\ free}$ 의 오차공분산은 다음과 같다.

$$\begin{aligned} P_{t(t_k|t_k)}^{bias\ free} &= E \left[\begin{pmatrix} X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^{bias\ free} \\ (X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^{bias\ free})^T \end{pmatrix} \right] \\ &= E \left[\begin{pmatrix} X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^b + G_{t(t_k)}^f \hat{\mathbf{b}}_{(t_k)} \\ (X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^b + G_{t(t_k)}^f \hat{\mathbf{b}}_{(t_k)})^T \end{pmatrix} \right] \end{aligned} \quad (37)$$

$$\begin{aligned} &= E \left[\begin{pmatrix} X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^b + G_{t(t_k)}^f (\mathbf{b}_{(t_k)} - \hat{\mathbf{b}}_{(t_k)}) \\ (X_{t(t_k)} - \hat{X}_{t(t_k|t_k)}^b + G_{t(t_k)}^f (\mathbf{b}_{(t_k)} - \hat{\mathbf{b}}_{(t_k)}))^T \end{pmatrix} \right] \\ &= E \left[\begin{pmatrix} \hat{e}_{t(t_k|t_k)} + G_{t(t_k)}^f \hat{e}_{t(t_k)}^b \\ (\hat{e}_{t(t_k|t_k)} + G_{t(t_k)}^f \hat{e}_{t(t_k)}^b)^T \end{pmatrix} \right] \end{aligned}$$

$$\hat{e}_{t(t_k)}^b = \mathbf{b}_{(t_k)} - \hat{\mathbf{b}}_{(t_k)} \quad (38)$$

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

$$P_{t(t_k|t_k)}^{bias\ free} = P_{t(t_k|t_k)} + G_{t(t_k)}^f \Sigma_{(t_k)}^b G_{t(t_k)}^{fT} \quad (39)$$

본 논문에서 제안하는 알고리즘의 전체적인 흐름도는 아래 그림과 같다.

Fig. 2. Bias estimation and compensation flowchart

융합트랙을 이용한 바이어스 추정 및 보상과정은 위 그림과 같이 i 번째 센서에서 표적 t 에 대한 측정치가 들어오면 융합트랙의 예측치를 이용하여 바이어스 의사 측정치를 생성하고 바이어스를 추정하며 융합트랙에서는 해당 측정치를 이용하여 추정치를 쇄신하게 된다. 최종적으로 융합트랙의 추정치와 추정된 바이어스를 이용하여 바이어스 보정을 수행한다.

7. Simulation

본 논문에서 제안한 바이어스 추정기법의 성능을 확인하기 위하여 첫 번째 시뮬레이션에서는 2차원 환경에서 표적의 상태변수가 위치, 속도를 가질 때 EXX 기법과 [11]에서 제안한 방법을 이용한다. 두 번째 시뮬레이션은 3차원 환경에서 표적의 상태변수가 위치, 속도, 가속도를 가질 때 그 성능을 비교하며 EXX 기법은 해당 환경에서 시뮬레이션이 불가능하기 때문에 [11] 방법만을 이용한다.

가. 2차원 환경 시뮬레이션

시뮬레이션 파라미터는 [10]과 동일하게 두며 표적의 상태변수는 $X_{t(t_k)} = [x_{t(t_k)} \ y_{t(t_k)} \ \dot{x}_{t(t_k)} \ \dot{y}_{t(t_k)}]^T$ 와 같고 동역학 모델과 공정잡음의 오차공분산은 Discretized Continuous White Noise Acceleration(DCWNA) 모델로 아래와 같다.

$$\Phi_{(t_{k+1}, t_k)} = \begin{bmatrix} I_2 & T I_2 \\ 0_2 & I_2 \end{bmatrix}, \quad T = t_{k+1} - t_k \quad (41)$$

$$Q_{(t_{k+1}, t_k)} = \begin{bmatrix} \frac{1}{3} T^3 I_2 & \frac{1}{2} T^2 I_2 \\ \frac{1}{2} T^2 I_2 & T I_2 \end{bmatrix} \tilde{q} \quad (42)$$

여기서 $\tilde{q}$ 는 power spectral density로 $6\text{m}^2/\text{s}^3$ 으로 두었다. 센서의 경우 두 개의 센서가 동일하게 1초의 주기를 가지지만 0.5sec의 시간 offset을 가지는 비동기 환경이다. 각각의 센서는 (0km, 0km)와 (10km, 0km)에 존재하며 두 센서의 측정잡음의 편차는 거리와 방위각 성분으로 $\sigma_1^r = \sigma_2^r = 10\text{m}$, $\sigma_1^\theta = \sigma_2^\theta = 1\text{mrad}$ 이며 각 센서의 바이어스는 거리, 방위각 방향의 offset bias와 거리, 방위각 방향의 scale bias 순으로 $\beta_1 = \beta_2 = [20\text{m}, 2\text{mrad}, 3 \times 10^{-5}, 0.2]^T$ 와 같다.

모든 센서의 바이어스 추정치는 0 벡터로 바이어스 오차공분산은 $\Sigma^b = \text{diag}[100^2, 0.2^2, 0.01^2, 0.1^2]$ 으로 초기화 하였다. 표적의 초기값은 알고 있다고 가정하며 실제 표적상태에 평균이 0이고 분산이 초기 오차공분산인 가우시안 분포로 초기화한다. 여기서 초기 오차공분산은 모든 표적에 대해서 $P_{t(0|0)} = \text{diag}[10^2, 10^2, 5^2, 5^2]$ 로 두었으며 또한 바이어스 행렬인 $G_{t(0)}^f$ 는 영행

렬로 둔다. 표적의 경우 총 32개의 표적이 x축과 y축 방향으로 각각 20m/s 속도로 이동하며 표적의 시나리오 오는 다음 그림과 같다.

Fig. 3. Simulation#1 - target scenario

몬테칼로 시뮬레이션은 총 100회를 수행하였으며 바이어스를 추정한 결과는 다음과 같다.

Fig. 4. Range offset bias RMSE

여기서 EXX가 EXX를 수행한 결과이며 Local Filter가 [11] 방법이고 SFT가 본 논문에서 제안한 방법을 이용한 결과를 의미한다. 위 결과를 비교해볼 때 본 논문에서 제안한 방법이 보다 빠르게 바이어스를 추정하고 추정성능도 향상된다는 것을 알 수 있다.

Fig. 5. Azimuth offset bias RMSE

Fig. 6. Range scale bias RMSE

Fig. 7. Azimuth scale bias RMSE

나. 3차원 환경 시뮬레이션

표적의 상태변수는 3차원의 위치, 속도, 가속도로

$$X_{t(t_k)} = [x_{t(t_k)} \ y_{t(t_k)} \ z_{t(t_k)} \ \dot{x}_{t(t_k)} \ \dot{y}_{t(t_k)} \ \dot{z}_{t(t_k)} \ \ddot{x}_{t(t_k)} \ \ddot{y}_{t(t_k)} \ \ddot{z}_{t(t_k)}]^T$$

와 같으며 동역학 모델과 공정잡음의 오차공분산 (DCWNA)은 다음과 같다.

$$\Phi_{(t_k+1, t_k)} = \begin{bmatrix} I_3 & TI_3 & T^2/2I_3 \\ 0_3 & I_3 & TI_3 \\ 0_3 & 0_3 & I_3 \end{bmatrix} \quad (43)$$

$$Q_{(t_k+1, t_k)} = \begin{bmatrix} \frac{T^5}{20}I_3 & \frac{T^4}{8}I_3 & \frac{T^3}{6}I_3 \\ \frac{T^4}{8}I_3 & \frac{T^3}{3}I_3 & \frac{T^2}{2}I_3 \\ \frac{T^3}{6}I_3 & \frac{T^2}{2}I_3 & TI_3 \end{bmatrix} \tilde{q} \quad (44)$$

$\tilde{q}$ 는 $4m^2/s^3$ 이며 센서는 총 3개를 이용한다. 첫 번째 센서는 시뮬레이션 시간 0.3sec에서 1sec 주기로 측정치를 획득하며 두 번째는 센서는 0.5sec, 세 번째 센서는 0.7sec에서 각각 1sec 주기로 측정치를 획득한다. 각 센서의 초기 위치는 ENU좌표계상에서 (0km, 0km, 0km), (100km, 0km, 0km), (-100km, 0km, 0km)에 존재하며 모든 센서의 측정잡음은 거리, 고각, 방위각순으로 (10m, 1mrad, 1mrad) 이다. 센서의 바이어스 중 먼저 offset bias는 거리, 고각, 방위각 방향으로 각각 (20m, 2mrad, 2mrad)으로 하며 scale bias는 $(3 \times 10^{-5}, 0.2, 0.2)$ 으로 모든 센서가 동일하다.

바이어스 추정치는 0으로 초기화하며 바이어스 오차 공분산은 $\Sigma^b = \text{diag}[100^2, 0.2^2, 0.2^2, 0.01^2, 0.1^2, 0.1^2]$ 으로 초기화 하였으며 각 트랙의 초기화는 2point differencing 방법을 이용한다. 표적의 경우 총 32개의 표적이 등속운동을 하다가 10~15sec 동안 1.5G로 등가속도운동을 한다. 모든 표적의 초기 속력은 212m/s이며 각 표적의 초기고도는 1km~22km으로 두었다. 다음은 표적의 궤적이다.

Fig. 8. Simulation#2 - target scenario

다음은 총 100회의 몬테칼로 시뮬레이션을 수행하여 얻은 바이어스를 추정 RMSE 결과다.

Fig. 9. Range offset bias RMSE

Fig. 10. Elevation offset bias RMSE

Fig. 11. Azimuth offset bias RMSE

Fig. 12. Range scale bias RMSE

Fig. 13. Elevation scale bias RMSE

Fig. 14. Azimuth scale bias RMSE

바이어스 추정결과를 통해 [11]과 본 논문에서 제안한 방법을 비교해볼 때 3차원 환경에서도 마찬가지로

제안한 방법이 보다 빠르게 바이어스를 추정하고 추정성능도 보다 향상된다는 것을 알 수 있다.

Fig. 15. Fusion track#8 position RMSE

Fig. 16. Fusion track#25 position RMSE

앞선 결과 그래프는 본 논문에서 제안한 바이어스 보상기법의 성능을 확인하기 위하여 Fig. 8과 같이 센서에서 가장 먼 융합트랙#8과 융합트랙#25의 바이어스 보상 유무에 따른 위치 RMSE를 비교한 것이다. 그 결과 바이어스 보상을 할 경우 바이어스 보상을 하지 않은 융합트랙에 비하여 위치추정 성능이 크게 향상되는 것을 볼 수 있다. 또한 바이어스로 인해 발생하는 오차의 크기는 거리와 각도에 비례하기 때문에 센서와 표적사이의 거리가 먼 융합트랙#8이 융합트랙#25 보다 위치 RMSE에 있어서 오차가 크게 나타나며 바이어스 보상에 있어서도 거리가 멀수록 보상되는 바이어스 값이 커지기 때문에 가까운 표적을 추적하는 융합트랙#25보다 융합트랙#8이 향상된 바이어스 보정 성능을 보이게 된다.

8. 결 론

본 논문에 제안한 바이어스 추정기법은 트랙의 추정성능에 영향을 받으며 [11]에서 제안한 로컬트랙보다 추정성능이 우수한 융합트랙을 이용하여 바이어스를 추정하기 때문에 보다 우수한 추정성능을 보이며 [11]과 다르게 융합을 수행하면서 실시간으로 바이어스를 추정하게 된다. EXX기법과 비교해 보면 센서의 개수와 표적의 상태변수 및 동역학 모델에 따라 알고리즘이 달라지지 않으며 측정치를 획득하게 되면 바로 바이어스를 채신 및 바이어스 보상이 이루어 질 수 있다는 장점을 가진다.

후 기

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

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.

- [6] YANYAN HU, ZHANSHENG DUAN, DONGHUA ZHOU, "Estimation Fusion with General Asynchronous Multi-Rate Sensors", IEEE Trans. Aerospace and Electronic Systems, Vol. 46, No. 4, pp. 2090~2102, 2010. 10.
- [7] Friedland, B., "Treatment of Bias in Recursive Filtering", IEEE Transactions on Automatic Control, AC-140, 1969. 8.
- [8] Ignagni, M. B., "An Alternate Derivation and Extension of Friendland's Two-Stage Kalman Estimator", IEEE Trans. on Automatic Control, AC-26, 3, 1981. 6.
- [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.
- [11] 김형원, 김도형, 박효달, 송택렬, "Multisensor Bias Estimation with Pseudo Measurement for Asynchronous Sensors", 한국군사과학기술학회지 제14권 제6호, pp. 1198~1206, 2011. 12.
- [12] Man Ying, Huang Jiang-yuan, Yu Zhi-hui, "3D Asynchronous Multisensor Target Tracking Performance with Bias Compensation", International Conference on Computer Engineering and Technology, Vol. 5, pp. 401~406, 2010. 4.