J. Korean Soc. Aeronaut. Space Sci. 50(9), 617-625(2022)
DOI:https://doi.org/10.5139/JKSAS.2022.50.9.617
ISSN 1225-1348(print), 2287-6871(online)

사격 차선 정렬을 위한 영상 기반의 관성 센서 편차 보상

아샤드 어웨이스[1], 박준우[2], 방효충[3], 김윤영[4], 김희수[5], 이용선[6], 최성호[7]

Vision Aided Inertial Sensor Bias Compensation for Firing Lane Alignment

Awais Arshad[1], Junwoo Park[2], Hyochoong Bang[3], Yun-young Kim[4], Heesu Kim[5],
Yongseon Lee[6] and Sungho Choi[7]
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea[1-3]
LIG Nex1, Seongnam, Republic of Korea[4,5]
Agency for Defense Development, Daejeon, Republic of Korea[6,7]

ABSTRACT

This study investigates the use of movable calibration target for gyroscopic and accelerometer bias compensation of inertial measurement units for firing lane alignment. Calibration source is detected with the help of vision sensor and its information in fused with other sensors on launcher for error correction. An algorithm is proposed and tested in simulation. It has been shown that it is possible to compensate sensor biases in firing launcher in few seconds by accurately estimating the location of calibration target in inertial frame of reference.

초 록

본 논문은 사격 차선 정렬을 위하여 움직일 수 있는 교정 대상을 이용해 각속도계와 가속도계의 편차를 보상하는 방법을 다룬다. 교정 대상에 대한 정보는 영상 센서를 통해 획득하며 이를 이용해 발사장치에 부착된 관성측정 장치의 오차를 보정한다. 시뮬레이션을 통해 제안한 알고리즘의 성능을 검증하였으며, 특히 관성 좌표계에서 교정 대상에 대한 위치 정보를 정확하게 획득함으로써 발사장치의 관성 센서 편차를 효과적으로 보상할 수 있음을 보인다.

Key Words : Inertial Measurement Unit(관성 측정 장치), Extended Kalman Filter(확장 칼만 필터), Error Compensation(오차 보상), Sensor Fusion(센서 융합), Lane Alignment(차선 정렬)

Ⅰ. Introduction

We have many advanced automatic ground-to-ground, ground-to-air, and air-to-air weapon systems, but, conventional soldier on ground with automatic guns in their hands are still very important in modern day warfare. Human-soldiers as well as robotic-soldiers on ground can be assisted with good quality sensor suite along with efficient firing lane alignment algorithms so they can hit hostile targets with accuracy. For these algorithms to work, we also need robust error compensation techniques to handle sensor error in our sensor suite. One of the important error to consider in attitude reference systems are biases in accelerometers and gyroscopes which accumulate with time in dynamic environments.

† Received : March 10, 2020 Revised : June 15, 2022 Accepted : June 22, 2022

[1,2] Graduate Student, [3] Professor, [4-7] Researcher

[3] Corresponding author, E-mail : hcbang@kaist.ac.kr, ORCID 0000-0001-6016-8102

Output performance of most inertial reference systems suffer from continuous degradation of bias errors in accelerometer and gyroscopic measurements. There are many classical approaches to compensate these errors, dated back to 1930s. One approach utilized in early attitude reference systems was attitude error elimination by vehicle acceleration data provided by external means[1,2]. External data was compared with onboard sensor data and the difference was contributed to gravity projections proportional to the platform tilt errors. Based upon these errors, platform gyroscopes were recalibrated to compensate bias errors. This continuous correction procedure prevented the accumulation of attitude errors at the output of inertial measurement system, but it did not directly targeted the compensation of bias errors at the sensor level. That correction came with the modern studies of Kalman Filter.

Kalman filtering[5] is usually preferred as a more general and powerful approach to aided inertial system design as it uses a series of erroneous observations over time to produce estimates of unknown variables. The state vector to be estimated is composed of output errors and then, if necessary, is augmented by inertial sensor biases[6,7]. Thus the phases of attitude correction and sensor error estimation are replaced by a single update procedure of the Kalman filter. Our system is comprised for a solder wearing smart helmet, carrying a launcher, having a calibration target and aiming a hostile target. We assumes that a soldier carries very accurate GPS receiver and other inertial measurement sensors in its helmet. Soldier also carries a firing launcher with its own sensors but sensors on launcher are relatively imprecise as compared to helmet sensors. We have also assumed a movable calibration target which can be perceived by vision sensors available on soldier helmet as well as launcher. Further sensor level details are provided in Fig. 2. We propose a sensor fusion algorithm which correct launcher's sensor bias while using soldier's helmet sensor suite and calibration target. Our particular contribution is proposing an algorithm which calibrate launcher sensors with the help of a calibration target where target's own location is unknown. Exact location of calibration target is found by another estimation scheme in which helmet sensor suite is involved. Just because of taking help from helmet sensor suite, our calibration target need not to be fixed in some space. Instead, our calibration target can be mobile and it can move along with launcher in the battle field. We show that a launcher can provide very accurate attitude information while carrying relatively imprecise sensors prone to biases and without any need for dedicated GPS receiver.

After introduction section, this paper presents system overview and define frames of reference in section II. System overview is followed by a section III which describes soldier attitude determination and localization in inertial frame of reference. This section also describes a strategy for the localization of calibration target in inertial frame of reference. We then present launcher error compensation algorithm in section IV. Results and Discussion section is presented in section V.

II. System Overview

Our system consists of a mobile soldier carrying a smart helmet and a firing launcher. Soldier's helmet contains highly accurate sensor suite which includes GPS Receiver, three axis accelerometers, three axis gyroscopes and three axis magnetometers. It also contains an accurate RGBD stereo depth camera on top of its helmet. Soldier's body frame of reference is fixed to soldier's helmet. All the sensors on soldier's helmet are assumed to be fixed and aligned with soldier body frame of reference. Launcher carried by soldier also contains its own sensor suite which is comprised of three axis accelerometers, three axis gyroscopes, three axis magnetometers and a camera. Launcher does not contain any GPS receiver of its own and will rely primarily on Soldier's GPS receiver. We assume that sensor suite fixed on top of launcher is not as accurate as helmet's sensor suite. Accelerometers and gyroscopes on launcher have their own biases and their errors accumulate with time. Moreover, launcher have to face more dynamic environment in its operations which can further deteriorate attitude information acquired from launcher's sensor suite. Therefore, it's important to periodically compensate bias errors in launchers for accurate attitude referencing. We assume an external calibration target as reference that can be perceived by the vision sensors on both helmet and launcher. That calibration target is used to compensate launcher bias errors. Though calibration target is assumed movable, it should be static at the time of calibration. For this vision aided inertial bias compensation to work, we need to define some frames of references.

Our system primarily use three frames of references named navigation reference frame (N), solder reference frame (B) and launcher reference frame (L). Navigational frame of reference is simple North-East-Down frame of reference where x-axis points towards north, y-axis points towards east, and z-axis completes right hand rule. Navigational frame is assumed inertial frame. Soldier's frame of reference is fixed with soldier's helmet. We have further assumed that all sensors on top of soldier's helmet are fixed and alighted with soldier's frame of reference (B). Therefore, there is no misalignment between helmet's sensor axes and soldier's helmet frame of reference and this is true for all sensors on top of helmet. Similarly, launcher's frame of reference (L) is fixed with launcher body and all sensor's on launcher are perfectly aligned with launcher's frame of reference (L). These frame of references are depicted in Fig. 1.

Lastly, we have two targets in our system named hostile target and calibration target. Hostile target needs to be aimed by launcher. Location of hostile target is assumed to be known in navigational frame of reference. To aim a static hostile target with accuracy, we need to accurately compensate inertial sensor bias errors with the help of efficient sensor fusion algorithm so we may achieve accurate attitude referencing. For that inertial sensor bias compensation we have an external calibration target. Moreover, both soldier's helmet and soldier's launcher must be static with calibration target in their field of view at the time of correction. Once bias correction is made, calibration target, soldier's helmet, and launcher can move for post-bias-correction firing lane alignment.

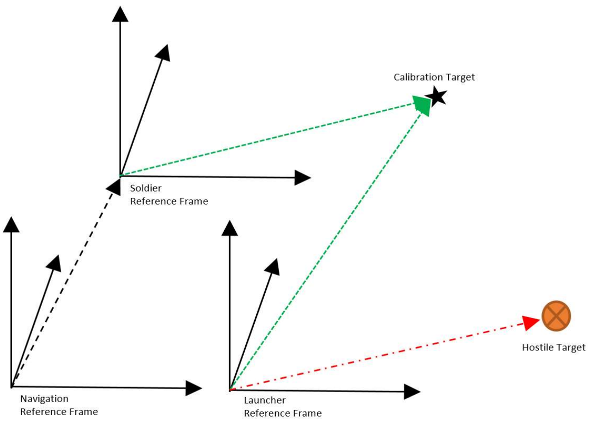

Fig. 1. Frames of references

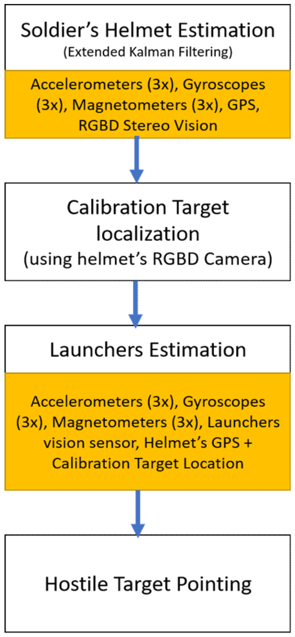
Soldier's Helmet Estimation
(Extended Kalman Filtering)
Accelerometers (3x), Gyroscopes (3x), Magnetometers (3x), GPS, RGBD Stereo Vision
Calibration Target localization
(using helmet's RGBD Camera)
Launchers Estimation
Accelerometers (3x), Gyroscopes (3x), Magnetometers (3x), Launchers vision sensor, Helmet's GPS + Calibration Target Location
Hostile Target Pointing

Fig. 2. Block Diagram of Algorithmic Flow

Flow of the algorithm is shown in Fig. 2. we start with attitude and location estimation of soldier's helmet in navigational frame of reference. we use accelerometers, gyroscopes, magnetometers and GPS receiver onboard the helmet for this task. Later we localize our calibration target with the help of RGBD camera on the helmet. we localize our calibartion target in both soldiers frame and navigation frame of reference. Once we successfully localize our calibration target in navigation frame, our main task of calibrating launcher starts. we use launcher's onboard camera to measure calibration target location and fuse that information with other sensors onboard the launcer with the help of Extended Kalman Filter. This helps us compensate launcher's sensor biases and aim the hostile target correctly.

Ⅲ. Soldier's Attitude and Localization

Before we can make any compensation for bias errors of launcher sensors, we need to accurately know the exact location of calibration target in navigation frame of reference because calibration target serves as an external reference for launcher's bias error correction. Calibration target has no sensor of its own and it is considered movable with respect to soldier and launcher (but immovable at the time of correction). Even though it's not possible to directly localize calibration target in navigational frame, we may localize calibration target indirectly with the help of soldier's helmet in navigational frame. We assume that our soldier carries an accurate depth sensor on its helmet which can be used to localize calibration target in soldier's frame of reference (B). On the other hand, soldier carry all necessary sensors on its helmet for its own localization as well as attitude determination in navigational frame of reference (N). Thus calibration target localization in soldier's frame of reference (B) can be transformed in navigational frame (N) with the help of soldier's sensor suite. Therefore we first need to localize and determine attitude of soldier's helmet.

In soldier attitude and localization step we need to find the attitude and position of soldier in navigational frame of reference (N). Navigational frame of reference is North-East-Down (NED) frame in our case. We make use of GPS, accelerometers, gyroscopes and magnetometer sensors in soldier's helmet to localize solder in navigational frame. Process starts with the definition of state transition matrix followed by process and measurement model definitions. Soldier attitude and localization is achieved by sensor fusion using Extended Kalman Filter[8,9].

3.1 State Transition

The state transition models form the core of the Extended Kalman Filter (EKF) prediction stage by performing system propagation from current time-step to the next time-step. State transition model uses high quality sensor inputs at current time-step to propagate system states in future. Process noise vectors relating current state to sensor noise are also defined in state transition matrices. These state matrices enable the computation of process covariance matrix, Q, and the process Jacobian, F. Both process covariance matrix, Q, and the process Jacobian, F, are used further in Extended Kalman Filter to propagate system covariance, P, from current time-step to future time-step. Our state vector is comprised of soldier position, soldier velocity, attitude quaternion, accelerometer bias and gyroscopes bias (1). Propagated system state is a function of system's previous state, system input and system process noise vector as shown in Equation (2).

$$\vec{x}=\begin{Bmatrix} \vec{r}^{N} \\ \vec{v}^{N} \\ {}^{N}\vec{q}^{B}_{bias} \\ \vec{\omega}^{B}_{bias} \\ \vec{a}^{B}_{bias} \end{Bmatrix}=\begin{Bmatrix} \text{NED Position}(3) \\ \text{NED Velocity}(3) \\ \text{Body Attitude}(4) \\ \text{Angular Rate Bias}(3) \\ \text{Accelerometer Bias}(3) \end{Bmatrix} \tag{1}$$

$$\vec{x}_k=\vec{f}(\vec{x}_{k-1},\vec{u}_{k-1})+\vec{w}_{k-1} \tag{2}$$

where $\vec{x}_k$ is the system state-vector, $\vec{u}_{k-1}$ is the input-vector (consisting of sensor signals) and $\vec{\omega}_{k-1}$ is the process noise vector. Process noise vector is a function of individual sensor noise vectors corresponding to the angular-rate and accelerometer biases. Expanded view of underling computations in state-transition vector $\vec{f}$ is shown in Equation (3) [11].

$$\vec{f}=\begin{Bmatrix} \vec{r}^{N}_{k-1}+\vec{v}^{N}_{k-1}\cdot dt \\ \vec{v}^{N}_{k-1}+\left[{}^{N}R^{B}_{k-1}\cdot Z_a-\vec{a}^{N}_{grav,k-1}\right] \\ \left[I_4+\frac{dt}{2}\cdot Z_g\right]\cdot{}^{N^{B}_{k-1}}\vec{q} \\ I_3 \\ I_3 \end{Bmatrix} \tag{3}$$

$$Z_a=\left(\vec{a}^{B}_{meas,k-1}-\hat{a}^{B}_{bias,k-1}\right)$$

$$Z_g=\left(\Omega_{meas,k-1}-\Omega_{bias,k-1}\right)$$

3.2 Process Model

As the state-transition model is nonlinear and difficult to compute, the state-transition vector cannot be directly used to propagate the covariance matrix, P, in Extended Kalman Filters. Therefore we linearize state-transition vector by Taylor-series expansion. Linearized solution is referred to as process Jacobian. Process Jacobian, F, along with process noise covariance matrix, Q, can be used to propagate covariance matrix, P, forward in time as shown in Equation (6). Process Jacobian, F, and process noise matrix, Q, are shown in Equations (4) and (5) respectively. The process covariance matrix serves the purpose of weighting matrix for the system process. It relates the covariance between individual elements of each process-noise vector [11].

$$F_{k-1}=\frac{\partial\vec{f}}{\partial\vec{x}_{|\vec{x}_{k-1},\vec{u}_{k-1}}} \tag{4}$$

$$Q_k=\begin{bmatrix} \Sigma_r & 0_3 & 0_{3\times4} & 0_3 & 0_3 \\ 0_3 & \Sigma_v & 0_{3\times4} & 0_3 & 0_3 \\ 0_{4\times3} & 0_{4\times3} & \Sigma_q & 0_{4\times3} & 0_{4\times3} \\ 0_3 & 0_3 & 0_{3\times4} & \Sigma_{\omega b} & 0_3 \\ 0_3 & 0_3 & 0_{3\times4} & 0_3 & \Sigma ab \end{bmatrix} \tag{5}$$

$$P_{k|k-1}=F_{k-1}P_{k-1|k-1}F^{T}_{k-1}+Q_{k-1} \tag{6}$$

3.3 Measurement Model

We need two kinds of measurements for Extended Kalman Filtering. First measurement is sensor based measurement $\vec{z}_k$ while the other measurement $\vec{h}_k$ is taken with the measurement model (7), (8). A measurement model relates the system states to the system measurements. It is possible to choose among various measurement models for a given EKF implementation. Our measurement vector $\vec{z}_k$ is comprised of position, velocity, and attitude information. Out of these three measurement values, only velocity information is directly available through our GPS receiver. Other two measurements need to be derived from available sensor data. On the other hand, our measurement model vector $\vec{h}_k$ have position and velocity information directly available from system state matrix. Attitude (roll, pitch and yaw) information in measurement model may be derived from Equation (9)-(11).

$$\vec{z}_k = \begin{Bmatrix} \vec{r}^{\,N}_{GPS} \\ \vec{v}^{\,N}_{GPS} \\ N^B_{meas}\vec{\Theta} \end{Bmatrix} \tag{7}$$

$$\vec{h}_k = \begin{Bmatrix} \vec{r}^{\,N}_{pred} \\ \vec{v}^{\,N}_{pred} \\ N^B_{pred}\vec{\Theta} \end{Bmatrix} \tag{8}$$

$$^{\perp}\phi^B_{pred} = \operatorname{atan2}\left(2(q_2q_3 + q_0q_1), (q_0^2 - q_1^2 - q_2^2 + q_3^2)\right) \tag{9}$$

$$^{\perp}\theta^B_{pred} = -\operatorname{asin}\left(2(q_1q_3 - q_0q_2)\right) \tag{10}$$

$$^{\perp}\psi^B_{pred} = \operatorname{atan2}\left(2(q_1q_2 + q_0q_3), q_0^2 + q_1^2 - q_2^2 - q_3^2\right) \tag{11}$$

3.4 Extended Kalman Filter for Soldier

Extended Kalman Filter operates in two steps namely prediction step and update step. The first step in EKF is prediction step in which system state vector is propagated in time along with system covariance estimate using sensor readings. Equations (12), (13) are used for update step where P, F, and Q represent process covariance, process Jacobian and process noise covariance respectively.

$$\vec{x}_{k|k-1} = f\left(\vec{x}_{k-1|k-1}, \vec{u}_{k|k-1}\right) \tag{12}$$

$$P_{k|k-1} = F_{k-1}P_{k-1|k-1}F^T_{k-1} + Q_{k-1} \tag{13}$$

The second step in EKF is update step in which we update our estimate considering difference between measurement vector $\vec{z}_k$ and measurement model $\vec{h}_k$. This difference is referred to as measurement error or innovation error (14).

$$\vec{v}_k = \vec{z}_k - \vec{h}_k \tag{14}$$

After we are done with state estimation, covariance prediction, and innovation error calculation, we can easily estimate state vector using Equations (15)-(20) Update step makes use of measurement Jacobian, H, which is obtained by taking partial derivatives of measurement model with respect to system states. Matrices 'S' and 'K' are called innovation covariance and Kalman gains respectively.

$$S_k = H_kP_{k|k-1}H^T_k + R_k \tag{15}$$

$$K_k = P_{k|k-1}H^T_kS^{-1}_k \tag{16}$$

$$\Delta\vec{x}_k = K_k\vec{v}_k \tag{17}$$

$$\vec{x}_{k|k} = \vec{x}_{k|k-1} + \Delta\vec{x}_k \tag{18}$$

$$\Delta P_k = -K_kH_kP_{k|k-1} \tag{19}$$

$$P_{k|k} = P_{k|k-1} + \Delta P_k \tag{20}$$

Once we localize soldier in navigational frame of reference, we may also localize calibration target in navigational frame of reference as we assume that our soldier has accurate RGBD sensor capable of locating calibration target. Since we have assumed that our system contain accurate RGBD sensor, we don't need a separate filter for noise cancellation. But if someone assumes a noisy RGBD sensor, another EKF may again be applied for accurate localization of calibration target in navigational frame. It should be noted that state vector for calibration target's location estimation is different from state vector for soldier localization in navigational NED frame of reference (N).

Ⅳ. Launcher Error Compensation

Once we determine the calibration target location in navigation frame of reference using sensor suite available on robot's helmet, we can use that target location to be detected with the help of vision sensor on launcher and fuse that information with other sensors on board the launcher. Launcher is not supposed to have a GPS receiver but it is assumed to be in closed proximity of soldier's helmet. Moreover, we assume that calibration target in not in close proximity of launcher. Because of these assumptions we can use the GPS measurements from robot/soldier's helmet and fuse that information with other sensors available on launcher. Launcher state vector can be represented as $\vec{x}_l^N = \left(\vec{r}_l^N, \vec{v}_l^N, \vec{q}_l^{NL}, \vec{\omega}_{l,bias}^N, \vec{a}_{l,bias}^N\right)^T$. Subscript l in state vector represents values related to launcher. This state-vector is different from state-vectors in section 3.1 because it represents states of the launcher instead of helmet. Methodology for sensor fusion from IMU and GPS is same as described for the earlier case of helmet while we need to use different measurement model. This time we shall use calibration target's visual information to compensate gyro bias and accelerometer bias in launcher frame of reference.

$$\vec{x} = \begin{Bmatrix} \vec{r}^N \\ \vec{v}^N \\ {}^{N}\vec{q}^{L} \\ \vec{\omega}_{l,bias}^N \\ \vec{a}_{l,bias}^N \end{Bmatrix} \tag{21}$$

State vector in Equation (21) is comprised of launcher position in navigational frame (N), launcher velocity in navigational frame, unit quaternion between launcher frame (L) and navigational frame, gyroscopic biases and accelerometer biases represented in navigational frames of references.

4.1 Measurement Model

While describing sensor error compensation for soldiers helmet, we went through State transition, process model, measurement model, and Extended Kalman Filtering Steps. When dealing with launchers error compensation, we note that steps are somewhat similar but major difference lies in measurement model. Measurement model for soldier's attitude and localization contained only position, velocity and Euler angles of soldier's helmet while measurement model $\vec{h}_k$ in launcher's error compensation algorithm contains an extra element $\vec{h}_{pred}^N$. Vector $\vec{h}_{pred}^N$ points from the camera's optical center position to the 3D point location of calibration target, expressed in navigational frame of reference (N). Sensor measurement vector $\vec{z}_{l,k}$ along with measurement vector from model $\vec{h}_{l,k}$ for launcher's error compensation algorithm is presented in Equations (22), (23).

$$\vec{z}_{l,k} = \begin{Bmatrix} \vec{r}_{GPS}^N \\ \vec{v}_{GPS}^N \\ \vec{\Theta}_{meas}^{NL} \\ \vec{h}_{meas}^N \end{Bmatrix} \tag{22}$$

$$\vec{h}_{l,k} = \begin{Bmatrix} \vec{r}_{pred}^N \\ \vec{v}_{pred}^N \\ \vec{\Theta}_{pred}^{NL} \\ \vec{h}_{pred}^N \end{Bmatrix} \tag{23}$$

To predict vector $\vec{h}_{pred}^N$ we use standard pinhole camera model which makes a visual measurement of a calibration target. By assuming simple pinhole camera model, we can use standard equation of pinhole camera to find a vector from launcher's camera optical center position to the 3D point location of calibration target. Since we know the locations of both camera optical center and calibration target, finding this vector is a simple algebraic operation. Once we calculate $\vec{h}_{pred}^N$ vector and measure similar vector through real camera, we can simply find the difference between these two vectors to input in Kalman Filter's equations. Hence, known location of calibration target can be used as an external reference to compensate launcher's biases.

4.1.1 Visual Measurement

We have assumed a standard pinhole camera which makes a visual measurement z_{uv} of a calibration target. This measurement is a sum of true measurement of pixel location for calibration target in image frame and Gaussian white noise.

$$z_{uv} = y_{uv} + v_{uv} \tag{24}$$

where $y_{uv} = [u_d, v_d]$ represents the true position in the image plane of the projection of calibration target. u_d and v_d are distorted pixel coordinates. The term v_{uv} represents the uncertainty associated with visual measurements and is modeled by a Gaussian white noise with power spectral density (PSD) σ_{uv}^2. Considering a central projection camera model, the image plane is located in front of the camera's origin upon which a non-inverted image is formed. The camera frame is assumed right-handed with the z-axis pointing to the field of view. The $\mathbb{R}^3 \Rightarrow \mathbb{R}^2$ projection of a 3D calibration target point located at $p^N=(x,y,z)^T$ to the image plane $p=(u,v)$ is defined by Equation (25), (26) where u and v are the coordinates of the image point p expressed in pixels units and (x',y',z') are coordinates of calibration target point in launcher's frame of reference. f is focal length of camera, while $(u_{0,}v_0)$ represent camera center-pixel location. $(u_{0,}v_0)$ also represent origin of camera frame of reference (C). Point p^C is the same 3D point p^N, but expressed in the camera frame. Inversely, a directional vector $h^C=\left[h^C_{x,}\, h^C_{y,}\, h^C_z\right]$ can be computed from the image point coordinates u and v using Equation (27).

$$u=\frac{x'}{z'}, v=\frac{y'}{z'} \tag{25}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f\ 0\ u_0 \\ 0\ f\ v_0 \\ 0\ 0\ 1 \end{bmatrix} p^C \tag{26}$$

$$h^C(u,v)=\left[\frac{u_0-u}{f}, \frac{v_0-v}{f}, 1\right] \tag{27}$$

The vector h^C points from the camera's optical center position to the 3D point location of calibration target. h^C can be expressed in the navigation frame by $\vec{h}^N_{pred}={}^NR^C h^C$, where ${}^NR^C$ is the camera to navigation rotation matrix. Similarly we may obtain $\vec{h}^N_{meas}$ simply by applying transformation ${}^NR^C$ to h^C_{meas}, where h^C_{meas} is measured pixel location of calibration target from camera fixed on top of launcher.

4.2 Extended Kalman Filter for Launcher

Operations of EKF for launcher error compensation are again similar to that of soldier's attitude and localization algorithm with a single difference in measurement Jacobian matrix H. Now the measurement Jacobian H is computed by taking the partial derivatives of launcher measurement model in Equation (23) with respect to launcher state vector elements. Once we calculate the measurement Jacobian, rest of process is same. We first apply filter prediction step using Equations (12), (13). We then calculate innovation error using Equation (14) while using measurement vector and measurement model from Equations (22) and (23) respectively. We then apply filter update step using Equations (15)-(20).

V. Results and Discussions

As discussed previously, the aim of this research work is to evaluate a working algorithm for firing lane alignment to help both human and robotic soldiers in battle field to aim accurately at hostile targets. Location of the hostile targets is assumed to be known beforehand in navigational frame of reference. As shown in Fig. 2, we first need to estimate location and attitude of helmet in navigational frame. Secondly, we need to localize calibration target and thirdly, we compensate launcher bias errors to aim hostile targets accurately. Our simulation work deal with first and third step only in which we try to compensate helmet's biases as well as launcher's biases only. In order to achieve this goal, our simulations used open source data set provided by Karlsruhe Institute of Technology (KITTI)[10]. This dataset provides 3D GPS/IMU data comprising location, speed, acceleration and meta-information acquired by sensors fixed on top of moving platform. This dataset provides both raw and processed data. We also added additional Gaussian noise in aforementioned dataset to check the validity of our algorithms. Thus processed data from KITTI dataset is assumed a ground truth and we have added a bias into this data. Our bias is modeled as a Gaussian random walk model consisting of a constant offset plus a random drift. We processed noisy data from our algorithm and evaluated it against noiseless data. We prepared two different versions of noisy data for each "soldier's attitude and localization algorithm" and "launcher error compensation algorithm".

In first case, we prepared noisy data for soldier's helmet. This dataset included GPS measurements as well as inertial measurement unit's (IMU) data. We processed this data from sensor fusion algorithm to see if we can correctly estimate biases for both accelerometers and gyroscopes. Since we had assumed that we have very accurate high quality sensors on soldier's helmet, bias errors were assumed very small. We assumed a measurement error with 0.00005rad/sec standard deviation of gyroscopic bias and 0.00001rad/sec of gyroscopic drift along all three axis. Similarly we assumed a measurement error with 0.00005m/sec^2 standard deviation of accelerometer bias and 0.00001m/sec^2 of accelerometer drift along all three axis. We observed in all cases that we were able to estimate and correct biases within one second and this bias correction persisted over a long period of time. Fig. 3 illustrates true and estimated biases in soldier's helmet gyroscope. Similarly Fig. 4 illustrates true and estimated biases in soldier's helmet accelerometers. States X_{11}, X_{12}, and X_{13} correspond to gyroscopic bias along x-axis, y-axis and z-axis respectively in rad/sec, while states X_{14}, X_{15}, and X_{16} corresponds to accelerometer bias along x, y, and z axis respectively in m/sec^2. We can see that estimated values of gyroscopic bias as well as accelerometer bias are pretty close to their true values of biases in Fig. 3 and Fig. 4. Once we compensate these estimated values of biases from measured values, we may get corrected values of orientation and location from our sensor suite. Soldier orientation and localization after bias compensation are presented in Fig. 5 and Fig. 6 respectively.

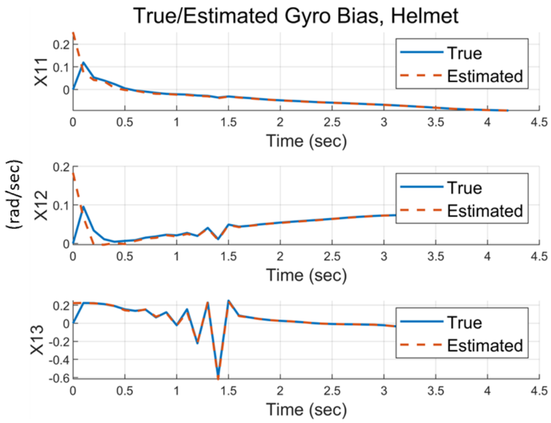

Fig. 3. Gyroscopic biases in soldier's helmet. X_{11}, X_{12}, and X_{13} represent biases along x-axis, y-axis and z-axis respectively with units in rad/sec

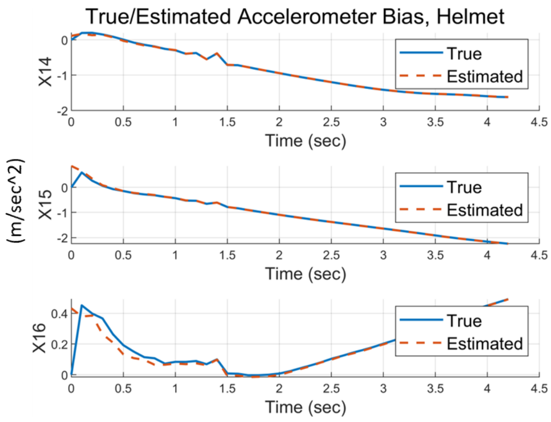

Fig. 4. Accelerometer biases in soldier's helmet. X_{14}, X_{15}, and X_{16} represent biases along x-axis, y-axis and z-axis respectively with units in m/sec^2 rad/sec

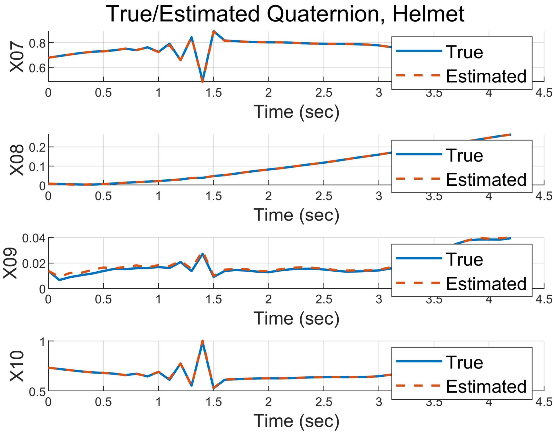

Fig. 5. Solder's helmet attitude quaternion with respect to navigational frame of reference

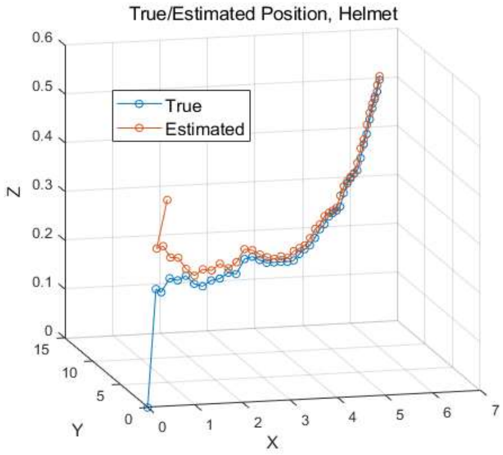

Fig. 6. Solder's localization in navigational frame. Units in meters

In the second case, we prepared noisy data for launcher. Both accelerometer and gyroscopic measurements are assumed to have measurement noise modeled as Gaussian and biases modeled as Gaussian random walks. Though launcher dataset included accelerometer, gyroscopic and magnetometer measurements, it did not include GPS measurement. Launcher used GPS measurements from soldier's helmet for its error compensations because of its close proximity with soldier. Amount of deliberate error induced in launcher's measurement dataset was considerably more than that of soldier-helmet's measurements presented before. It is evident from results in Fig. 6 that gyroscopic biases along all axes are estimated quite effectively within few seconds. States X_{11}, X_{12}, and X_{13} in Fig. 7 correspond to gyroscopic bias along x, y, and z axis respectively with units in rad/sec. Though it takes around four

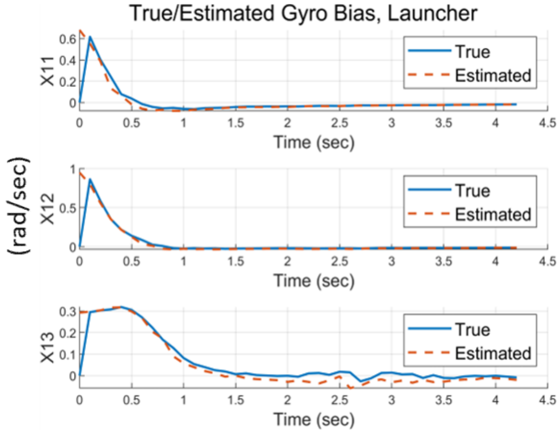

Fig. 7. Gyroscopic biases in soldier's launcher. X_{11}, X_{12}, and X_{13} represent biases along x-axis, y-axis and z-axis respectively with units in rad/sec

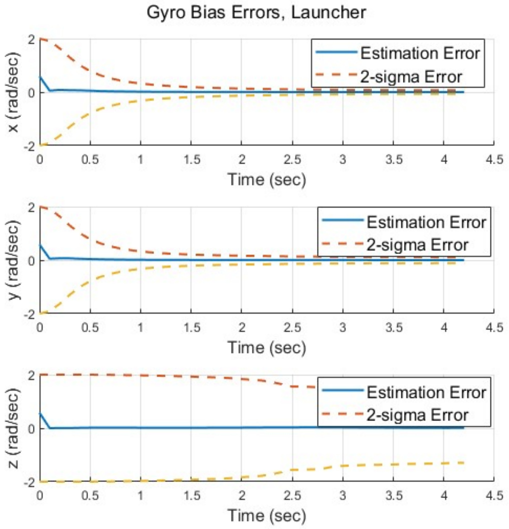

Fig. 8. Gyroscopic bias Errors in soldier's launcher along x-axis, y-axis and z-axis respectively with units in rad/sec

senconds to correctly estimate bias along z-axis for our proposed algorithm, estimation of biases along other two axis takes below one second.

Lastly, we present error plots for gyro bias shown in Fig. 8. along with 2-sigma of bias error. we may see from plots in Fig. 8. that error as well as its 2-sigma values reduce with time.

Acknowledgement

This research was supported by LIG Nex1 as a part of the research project under the contract number LIGNEX1-2018-0362.

References

1) Glitscher, K., “Indicator,” U.S. Patent No. 1932210, 24 October 1933.

2) Rafferty, C. A., “Vertical gyro erection system,” U.S. Patent No. 3,285,077. 15 November 1966.

3) Shaw, J. C., Gilbert, J. F., Olbrechts, G. R. and McIntyre, M. D., “Integrated Strapdown Air Data Sensor System,” U.S. Patent No. 4303978, 1 December 1981.

4) Salychev, O., *Applied Inertial Navigation: Problems and Solutions,* Moscow, Russia, BMSTU press, 2004.

5) Kalman, R. E., “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME - Journal of Basic Engineering,* Vol. 82, No. 1, 1960, pp. 35~45

6) Farrell, J. and Barth, M., *The Global Positioning System and Inertial Navigation,* New York, NY, USA, Mcgraw-Hill, 1999.

7) Grewal, M. S., Weill, L. R. and Andrews, A. P., *Global Positioning Systems, Inertial Navigation, and Integration,* John Wiley & Sons, 2007.

8) Grewal, M. S. and Andrews, A. P., *Kalman filtering: Theory and Practice with MATLAB,* John Wiley & Sons, 2014.

9) Sabatini, A. M., “Quaternion-based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing,” *IEEE transactions on Biomedical Engineering,* Vol. 53, No. 7, 2006, pp. 1346~1356.

10) Geiger, A., Lenz, P., Stiller, C. and Urtasun, R., “Vision Meets Robotics: The KITTI Dataset,” *The International Journal of Robotics Research,* Vol. 32, No. 11, 2013, pp. 1231~1237.

11) Studio, A. N., *OpenIMU Developer Manual: State Transition Models,* 5 February 2020.