Estimation of states with data under Colored Measurement Noise (CMN)

[1]ELI G. PALE-RAMON, [1]YURIY S. SHMALIY, [2]LUIS J. MORALES-MENDOZA,
[2]MARIO GONZALEZ-LEE, [1]JORGE A. ORTEGA-CONTRERAS, [1]KAREN URIBE-MURCIA

[1]Electronics Engineering Department Universidad de Guanajuato, Salamanca, MEXICO
[2]Electronics Engineering Department Universidad Veracruzana, Poza Rica, MEXICO

Abstract: Object tracking is an area of study of great interest to various researchers, where the main objective is to improve estimation of the trajectory of a moving object. This is due to the fact that in the object tracking process there are usually variations between the true position of the moving object and the estimated position, that is, the object is not exactly followed throughout its trajectory. These variations can be thought of as Colored Measurement Noise (CMN) caused by the object and the movement of the camera frame. In this paper, we treat such differences as Gauss-Markov colored measurement noise.We use Finite Impulse Response and Kalman Filters with a recursive strategy on the tracking: predict and update. To demonstrate the filter with the best performance, tests were carried out with simulated trajectories and with benchmarks from a database available online. The UFIR modified for CMN algorithm showed favorable results with high precision and accuracy in the object tracking process with benchamark data and under no ideal conditions.While KF CMN showed better results in tests with simulated data under ideal conditions.

Keywords: Colored measurement noise, Bounding box, estimation, object tracking, Unbiased FIR filters, Kalman filter, Precision, F-score.

Received: July 17, 2021. Revised: May 29, 2022. Accepted: June 21, 2022. Published: August 3, 2022.

1. Introduction

Object tracking is a topic of great importance in the field of computer vision, since its applications range from security to autonomous vehicles. Object tracking has many practical applications such as image processing, robotics, industrial inspection, among others [1]–[3].For this reason, there is great interest from various researchers to develop algorithms that help improve object tracking processes.

The purpose of the object tracking process is to correctly estimate the position of a moving object throughout its trajectory in a scene. However, there are several factors that affect the performance of an algorithm in the tracking process, factors specific to the scene where the object is moving, such as lighting and background clutter or camera movements. That is, the target object is not followed exactly during the entire trajectory, there may be variations between the real position and the estimated position. These differences or variations can be considered as coloured measurement noise (CMN) [4]. Currently there is a great diversity of contributions to the area and it has become a challenging task to reduce estimation errors, but object tracking is still, but there is no single approach that provides the best solution to all the factors affecting the tracking task. That is why it remains a topic of great research interest.

One approach to address estimation problems in object tracking is the use of motion model and state estimators (MMS) as a method to avoid large tracking errors. Various investigations have shown that the MMS application provides high precision in position estimations if the state-space model of the moving object is correctly specified [5]–[10].

In this paper, we consider the frame variations as CMN, and apply the filters Standard Kalman (KF),modified Kalman for CMN(KF CMN, standard Unbiased Finite Impulse Response (UFIR), and modified UFIR for CMN (UFIR CMN)for estimate the state in object tracking process. The Filter test was performed with simulated sequences setting different conditions and benchmark data available on [11].To estimate the trajectories and to be able to evaluate the performance of the algorithms, the bounding box coordinates were used as data for the state estimation. Each position of the object in the trajectory is represented with a bounding box.

According to results, UFIR CMN and the KF CMN presented favorable results on simulated data with ideal conditions and known process and data noise. However with no ideal conditions noise values and with the the highest colored factor Ψ the UFIR CMN showed better results than KF CMN and the standard filter (KF, UFIR). Regarding the results obtained with benchmark data, the UFIR CMN presented the best performance followed closely by KF CMN.

This research was supported by the Mexican CONACYT-SEP Project A1-S-10287, Funding CB2017-2018

2. Object Tracking

Object tracking is a topic of great importance in the area of computer vision due to its wide field of application. Image processing operations seek the best recognition of objects in the track, which involves finding the right features to differentiate the target from other objects and the background of the scene. The content of an image can be described through its properties. To do this, it is necessary to calculate the properties of an image or region and use them as a basis for further classification. Therefore, shape parameter extraction is necessary for image representations. There are different ways to locate a target object, for example, its contour or finding the pixels of the object, one of the most used shape parameters in object tracking is the bounding box [12]. In addition, there is a diversity of databases that provide paths annotated with bounding boxes [11], [13]–[16].

2.1. Bounding Box

The bounding box (BB) is used to represent the target object during an entire trajectory. The bounding box can be defined as a rectangular box that encloses all target objects in a scene. Information about the position of objects in is contained in an array of bounding boxes. This information can be represented by the coordinates of the upper left and lower right corners of the bounding box [17]. The matrix consists of 4 columns and n rows, the number of rows correspond to the total number of detections, while the columns represent the dimensions of the bounding boxes: coordinate ”x”, coordinate ”y”, width (xw), height (yw).

In the tracking process, the ”x” and ”y” coordinate information of each bounding box is used to estimate the position of the object during the entire trajectory. However, there may be errors in the position estimation. To reduce these errors, an effective method is the application of a filtering method. A filtering method is used to predict the coordinates of a bounding box point. The estimation method consists of 2 stages: prediction and correction, whose objective is to mitigate the noise present in the object tracking process. The prediction indicates the next position of the bounding box based on its previous position. The update is a correction step, which includes the new measurement of the follow-up model and helps to improve the estimate of the filteringl [18], [19].

Bounding boxes are useful in object tracking algorithms when intersecting objects must be processed [20].In other words, the overlapping of the predicted bounding box (PBB) with the true bounding box (TBB) that contains the information of the target object along its trajectory in a sequence. Taking this into account, the performance of the tracking algorithm can be analyzed through the information provided by the bounding boxes. The most common ways to evaluate the performance of an object tracking algorithm, using the information provided by the bounding box matrix, are the precision and the F1-score.

3. Performance Metrics of Tracking Algorithm

The evaluation of the performance of the tracking algorithms can be done using the standard metrics, precision and accuracy, for calculate the accuracy we use the metric F-score. Precision can be defined as the percentage of the number of correct predictions over the total number of predictions. F-score, is a metric that combines precision and recall, this last term is calculated as the number of correct detected objects divided by the total number of detections in the ground truth [21]–[25].

3.1. Precision

To calculate the precision it is necessary to first calculate another metric, intersection over union (IoU), which indicates the percentage of overloap of the predicted bounding box over the True Bounding box (TBB). The variables used in the calculation of the precision are obtained from the comparison of the IoU result with an established threshold. The variables used for computed the precision are obtained from the comparison of the IoU result with an established threshold [21]–[23], [26]. The equations for calculating IoU and precision are (1) and (2), respectively.

$$IoU = \frac{IA}{(TBB - EBB) - IA} \tag{1}$$

$$Precision = \frac{\Sigma TP}{\Sigma TP + \Sigma FP} = \frac{\Sigma TP}{All \quad detections} \tag{2}$$

Where the IA is the area of intersection between the bounding box of the target object, the true bounding box (TBB), and the estimated bounding box (EBB). The TP is true positive, and FP is false positive.

The IoU metric allows establishing the degree or percentage of EBB overloap over TBB, for which it is necessary to establish an IoU threshold that works as the comparison parameter to establish whether it is a correct or incorrect detection. Generally, the IoU threshold is set to 0.5.

Considering a single object tracking, many measures to evaluate the performance of the tracking algorithm are based on the overlap comparison of the EBB versus the TBB. The possible qualification of the bounding box overloap in object tracking compared to a given threshold is shown below [21]–[23]:

- True Positive (TP). It is a correct detection of a bounding box, that is, the IoU between the EBB and TBB is greater than or equal to the established threshold value.
- False positive (FP). It is an incorrect detection of an object or an off-site detection. The IoU is less than the given threshold value, but greater than zero
- False negative (FN). It is an undetected TBB.

3.2. F-score

Most benchmarks use an axis-aligned bounding box as the Ground Truth and estimate the accuracy commonly by the IoU criterion between the EBB and the TBB, i.e. based on the comparison of the EBB versus the ground truth . One of the position-based evaluation parameters to measure accuracy is F-measure or F-score. Which is defined as the harmonic mean of the precision and recall. Where, Recall can be calculated as the number of correct detected objects divided by the total number of frames in the ground truth [21]–[23]. Recall is computed as shown below:

$$Recall = \frac{TP}{All \quad ground \quad truths} \tag{3}$$

F-score metric is delimited to the interval [0,1], the value will be 0 when the value of precision and recall is 0 and 1 when both values are equal to 1. Remembering that F-score considers both precision and recall [22], one way to calculate is given by:

$$F - score = 2\frac{Precsion \times Recall}{Precision + Recall} \tag{4}$$

Another way to calculate f-score directly considering the overloap qualifications between the EBB and the TBB given a given threshold value is computed by [23]:

$$F - score = \frac{TP}{TP + \dfrac{TP}{FN + FP}} \tag{5}$$

If we consider the total number of frames of the target's trajectory, the F-score result in both equations (4) and (5) would consist of a sum of its values in each frame divided by the total number of frames.

The tracking evaluation parameters, F-score and precision, are calculated concerning a threshold value. Which ranges from 0 to 1. Depending on the degree of overloap that we want to achieve, it will be the IoU comparison parameter, generally being a value of 0.5, that is, the algorithm is expected to be able to identify at least 50% of the area of the TBB. Also, the scale from 0 to 1 can be used and in this way evaluate at which IoU level the algorithm performs best.

4. Moving Object State-space Model

We consider a moving object with observation corrupted by CMN can be represented in discrete-time state-space using with the following state and observation equations:

$$x_n = A_n x_{n-1} + B_n w_n \,, \tag{6}$$

$$v_n = \Psi_n v_{n-1} + \xi_n \,, \tag{7}$$

$$y_n = C_n x_n + v_n \,, \tag{8}$$

where $x_n \in \mathbb{R}^K$ is the state vector, $y_n \in \mathbb{R}^M$ is the observation vector, $v_n \in \mathbb{R}^M$ is the colored Gauss-Markov noise, and $A_n \in \mathbb{R}^{K\times K}$ is the state transition matrix, $B_n \in \mathbb{R}^{K\times P}$ is the gain matrix model, $C_n \in \mathbb{R}^{M\times K}$ is the measurement matrix. Measurement noise transfer matrix Ψ_n is chosen such that the colored noise v_n remains stationary. The zero mean Gaussian noise vectors $w_n \sim \mathcal{N}(0, Q_n) \in \mathbb{R}^P$ and $\xi_n \sim \mathcal{N}(0, R_n) \in \mathbb{R}^M$ have the covariances Q_n and R_n and the property $E\{w_n \xi_k^T\} = 0$ for all n and k. We will consider the following estimates: the prior estimate $\hat{x}_n^- \triangleq \hat{x}_{n|n-1}$, posterior estimate $\hat{x}_n \triangleq \hat{x}_{n|n}$, prior estimation error $\epsilon_n^- = x_n - \hat{x}_n^-$, the posterior estimation error $\epsilon_n = x_n - \hat{x}_n$, the prior error covariance $P_n^- \triangleq P_{n|n-1} = E\{\epsilon_n^- \epsilon_n^{-T}\}$, and the posterior error covariance $P_n \triangleq P_{n|n} = E\{\epsilon_n {\epsilon_n}^T\}$.

Considering the motion of a physical system in space, the next position of the object can be calculated using Newton's equation of motion [27]:

$$cx = cx_0 + v_0\tau + \frac{1}{2}act^2 \,, \tag{9}$$

where cx is the position on the x-axis of the object position, cx_0 is the object's initial position, v_0 is the object's initial velocity, ac is the object's acceleration, and τ is the time interval.

Since the state estimation will be applied for the 4 coordinates of the bounding box, it is necessary to define the equations of motion for the other 3 bounding box measurements that make up the bounding box; which are the coordinates on the y-axis, the width of bounding box and height of bounding box.

Considering a constant acceleration model [28] , for the moving object state space model, the state transition (A) is block diagonal matrix with:

$$\begin{bmatrix} 1 & \tau & \dfrac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} , \tag{10}$$

where the block is repeated for the ''x'',''y'', ''width'', and ''height'' spatial dimensions. τ is the sample time.

The gain matrix model (B) and observation matrix (C) are defined as shown below:

$$B = \begin{bmatrix} \dfrac{\tau^2}{2} & 0 & 0 & 0 \\ \tau & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \dfrac{\tau^2}{2} & 0 & 0 \\ 0 & \tau & 0 & 0 \\ 0 & 0 & \dfrac{\tau^2}{2} & 0 \\ 0 & 0 & \tau & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dfrac{\tau^2}{2} \\ 0 & 0 & 0 & \tau \\ 0 & 0 & 0 & 1 \end{bmatrix} , \tag{11}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} . \tag{12}$$

4.1. Avoid the CMN

To avoid the CMN v_n in y_n is necesary the model modification using measurement differencing. For which, it´s consider a new observation z_n as a measurement difference.

$$\begin{aligned} z_n &= y_n - \Psi_n y_{n-1} , \\ &= C_n x_n + v_n - \Psi_n H_{n-1} x_{n-1} - \Psi_n v_{n-1} , \end{aligned} \tag{13}$$

and transform (4) to

$$z_n = D_n x_n + \bar{v}_n , \tag{14}$$

where $D_n = H_n - \Gamma_n$, $\Gamma_n = \Psi_n H_{n-1} F_n^{-1}$, $\bar{v}_n = \Gamma_n B_n w_n + \xi_n$, noise $\bar{v}_n$ is now white with the properties

$$\bar{R}_n = E\{\bar{v}_n \bar{v}_n^T\} = \Gamma_n \Phi_n + R_n , \tag{15}$$

$$E\{\bar{v}_n w_n^T\} = \Gamma_n B_n Q_n , \tag{16}$$

where $\Phi_n = B_n Q_n B_n^T \Gamma_n^T$, and model (1) and (5) has thus two white and time-correlated noise sources w_n and $\bar{v}_n$.

5. Standard and Modified for CMN Kalman Filter

The Kalman filter uses the equation of state of the linear system to estimate the state of the system through observation of input and output. Where the state is assumed to be distributed by a white Gaussian noise with zero mean. The KF consist of two steps, prediction and update and it is a recursive estimator [29], this means the previous estimated state need to be combined with new observations to calculate the best estimate of the current state. The KF requires knowledge of the system parameters, initial values, and measurement sequences.

The KF can estimate the state dynamics of the system iteratively [30], as already mentioned, consists of two steps: predict, where the optimal state $\hat{x}_n^-$ previous to observing y_n is calculated and update, where after observing y_n the optimal posterior state $\hat{x}_n$ is calculated. Additionally, it computes the prior estimation error $\epsilon_n^- = x_n - \hat{x}_n^-$, the posterior estimation $\epsilon_n^- = x_n - \hat{x}_n^-$, a priori estimate error covariance $P_n^- = E\{\epsilon_n^- \epsilon_n^{-T}\}$, and posterior estimate error covariance $P_n = E\{\epsilon_n \epsilon_n^T\}$.

5.1. Standard Kalman Filter

In the standard Kalman and the KF modified for CMN filters, in predicted phase is produced the a priori error covariance. Assuming that the process noise w_n is white Gaussian with zero mean, the prior state estimate is computed by (17), and the prior error covariance matrix is estimated by (18).

$$\hat{x}_n^- = A\hat{x}_{n-1} + B_n w_n , \tag{17}$$

$$P_n^- = A_n P_n A + B_n Q_n B_n^T , \tag{18}$$

Then, in the update phase, the current prior predictions are combined with the current state observation to redefine the state estimate and the error covariance matrix. The combination of the prediction with the current observation is used to obtain the optimal state estimate, and is called the posterior state estimate. The measurement y_n is corrupted by colored measurement noise v_n. The measurement residual is (34).

$$y_n = C\hat{x}_{n-1} + \bar{v}_n \tag{19}$$

The residual covariance matrix is obtained as follow:

$$S_n = C_n P_n^- C_n^T + R_n \tag{20}$$

The optimal Gain for Kalman is given by:

$$K_n = P_N^- C_n^T S_n^{-1} \tag{21}$$

A posteriori state estimate:

$$\hat{x}_n = \hat{x}_n^- + K_n(z_n - c\hat{x}_n^-) \tag{22}$$

A posteriori matrix of error covariance:

$$P_n = (I - K_n C)P_n^- \tag{23}$$

A pseudo code of standard Kalman filter is listed as Algorithm 1.

Algorithm 1: Standard Kalman Filter

```
Data: $y_n, \hat{x}_0, P_0, Q_n, R_n, u_n$
Result: $\hat{\mathbf{x}}_n$, $P_n$
begin
    for $n = 1, 2, \cdots$ do
        $\hat{x}_n^- = A\hat{x}_{n-1} + B_N u_n$
        $\bar{P}_n^- = A_n P_{n-1} A_n^T + B_n Q_n B_n^T$
        $S_n = C_n P_n^- C_n^T + R_n$
        $K_n = P_n^- C_n^T S_n^{-1}$
        $\hat{x}_n = \hat{x}_n^- + K_n z_n$
        $P_n = (I - K_n C_n) P_n^-$
    end for
end
```

5.2. Kalman Filter for CMN

The design of KF algorithms assuming CMN is deriving a new bias correction gain and error covariance for correlated noise. The measurement residual s_n transformed to

$$\begin{aligned} s_n &= z_n - D_n \hat{x}_n^- \\ &= D_n A_n \epsilon_{n-1} + D_n B_n w_n + \bar{v}_n , \end{aligned} \tag{24}$$

the innovation covariance S_n is given by

$$\begin{aligned} S_n &= E\{s_n s_n^T\} \\ &= D_n P_n^- D_n^T + R_n + C_n \Phi_n + \Phi_n^T D_n^T , \end{aligned} \tag{25}$$

and the estimation of KF for CMN is

$$\begin{aligned} \hat{x}_n &= \hat{x}_n^- + K_n s_n \\ &= A_n \hat{x}_{n-1} + K_n(z_n - D_n A_n \hat{x}_{n-1}) , \end{aligned} \tag{26}$$

where K_n is the bias correction gain that should be optimized for correlated w_n and $\bar{v}_n$. the estimation error ϵ_n can be written as

$$\begin{aligned} \epsilon_n &= x_n - \hat{x}_n \\ &= (I - K_n D_n) A_n \epsilon_{n-1} \\ &\quad + (I - K_n D_n) B_n w_n - K_n \bar{v}_n \end{aligned} \tag{27}$$

and the error covariance $P_n = E\{\epsilon_n \epsilon_n^T\}$ transformed to

$$\begin{aligned} P_n &= P_n^- - (P_n^- D_n^T + \Phi_n)K_n^T - K_n(P_n^- D_n^T + \Phi_n)^T \\ &\quad + K_n S_n K_n^T \,, \end{aligned} \tag{28}$$

where P_n^- is given by (18) and S_n by (24). The optimal gain K_n is given by

$$K_n = (P_n^- D_n^T + \Phi_n)S_n^{-1} \tag{29}$$

and (28) becomes

$$P_n = P_n^- - K_n(D_n P_n^- + \Phi_n^T)\,. \tag{30}$$

A pseudo code of the KF algorithm for CMN with correlated w_n and $\bar{v}_n$ is listed as Algorithm 2. It can be observed, if the value $\Psi_n = 0$, the algorithm becomes the standard Kalman Filter algorithm. and it becomes the standard KF by .

Algorithm 2: KF for CMN and Correlated w_n and $\bar{v}_n$

```
Data: $\mathbf{y}_n$, $\hat{x}_0$, $\mathbf{P}_0$, $\mathbf{Q}_n$, $\mathbf{R}_n$
Result: $\hat{\mathbf{x}}_n$, $P_n$
begin
    for $n = 1, 2, \cdots$ do
        $D_n = H_n - \Psi_n C_{n-1} A_n^{-1}$
        $z_n = y_n - \Psi_n y_{n-1}$
        $P_n^- = A_n P_{n-1} A_n^T + B_n Q_n B_n^T$
        $S_n = D_n P_n^- D_n^T + R_n + C_n \Phi_n + \Phi_n^T D_n^T$
        $K_n = (P_n^- D_n^T + \Phi_n) S_n^{-1}$
        $\hat{x}_n^- = A_n \hat{x}_{n-1}$
        $\hat{x}_n = \hat{x}_n^- + K_n(z_n - D_n \hat{x}_n^-)$
        $P_n = (I - K_n D_n) P_n^- - K_n \Phi_n^T$
    end for
end
```

6. Standard and Modified for CMN Unbiased Finite Impulse Response Filter

6.1. Standard Unbiased Finite Impulse Response Filter

Unlike the Kalman Filter, the Unbiased FIR does not require any information about initial conditions, the initial state, error covariance matrix P, and statistical noises Q and R, except for the zero mean assumption [8], [31], [31], [32]. UFIR is also an option for estimating states with incomplete measurement Information [33].

Instead, it requires an optimal averaging horizon [m, n] to minimize the mean square error(MSE), that is, the UFIR filter operates at once with N measurements on a horizon $[m, k]$ from $m = k - N + 1$. The UFIR filter cannot ignore the CMN v_n, which violates the zero mean assumption on short horizons [31].

Since the UFIR algorithm does not require noise statistics, the prediction phase calculates only one value, a priori state6. In the update step, the state estimate is combined with the actual observation state to refine the state The estimate is iteratively updated to the a posteriori state estimate using generalized noise power gain, measurement residual,and UFIR gain as shown in a pseudo-code of the UFIR algorithm is listed as 3. To initialize iterations, the algorithm requires a short measurement vector $y_(m.k) = [y_m \cdots y_k]^T$ and matrix (31):

$$H_{m,s} = \begin{bmatrix} C_m (A_s \ldots A_{m+1})^{-1} \\ C_{m+1}(A_s \ldots A_{m+2})^{-1} \\ \vdots \\ C_{s-1} A_s^{-1} \\ C_s \end{bmatrix}. \tag{31}$$

Algorithm 3: Standard UFIR Filter

```
Data: $y_n$, $u_n$, $N$
Result: $\hat{\mathbf{x}}_n$
begin
    for $n = N-1, N, \cdots$ do
        $m = n - N + 1$, $s = m - K + 1$
        $G_s = H_{m,s}^T H_{m,s}$
        $\bar{x}_s = G_s H_{m,s}^T (y_{m,s} - L_{m,s} u_{m,s}) + S_{m,s}^k u_{m,s}$
        for $l = s + k, \cdots$ do
            $\bar{x}_l^- = A \hat{x}_{l-1} + Bu$
              $G_l = [C_l^T C_l + (A_l G_{l-1} A_l^T)^{-1}]^{-1}$
              $Gain_l = G_l C_l^T$
              $\bar{x}_l = \hat{x}_l^- + Gain_l (y_l - C \bar{x}_l^-)$
        end for
        $\hat{x}_n = \bar{x}_n$
    end for
end
```

Where $S_{m,s}$ and $L_{m,s}$ are given by (33) and (32) respectivily, $S_{m,s}^k$ is the Kth row vector in (33)

$$L_{m,s} = diag(C_{m,s}) S_{m,s} \tag{32}$$

$$H_{m,s} = \begin{bmatrix} B_m & 0 & \cdots & 0 & 0 \\ A_{m+1} B_m & B_{m+1} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{s-1}^{m+1} B_m & A_{s-1}^{m+2} B_{m+1} & \cdots & B_{s-1} & 0 \\ A_{s-1}^{m+1} B_m & A_{s-1}^{m+2} B_m & \cdots & A_m B_{s-1} & B_s \end{bmatrix}. \tag{33}$$

6.2. Unbiased Finite Impulse Response Filter Modified for CMN

Since colored measurement noise is present in the data needed for the estimation, the following modified UFIR algorithm pseudocode was used to avoid the CMN, as shown in Algorithm 4. In the same way as with the standard UFIR algorithm, to initialize the iterations, the algorithm requires a short measurement vector $Y_{m,s} = [\, y_m \ \ldots \ y_s \,]^T$ and matrix

$$H_{m,s} = \begin{bmatrix} D_m (F_s \ldots F_{m+1})^{-1} \\ \vdots \\ D_{s-1} F_s^{-1} \\ D_s \end{bmatrix}. \tag{34}$$

It also follows that, by $\Psi_n = 0$, the Algorithms 4 and 2 becomes the standard UFIR filter and the standard Kalman filter, respectively [31].

Algorithm 4: UFIR Filter for CMN

```
Data: $N$, $y_n$
Result: $\hat{x}_n$
begin
    for $n = N-1, N, \cdots$ do
        $m = n - N + 1$ , $s = n - N + K$
        $G_s = (C_{m,s}^T C_{m,s})^{-1}$
        $\bar{x}_s = G_s C_{m,s}^T Y_{m,s}$
        for $l = s+1 : n$ do
            $D_l = H_l - \Psi_l H_{l-1} F_l^{-1}$
            $z_l = y_l - \Psi_l y_{l-1}$
            $G_l = [D_l^T D_l + (F_l G_{l-1} F_l^T)^{-1}]^{-1}$
            $K_l = G_l D_l^T$
            $\bar{x}_l^- = F_l \bar{x}_{l-1}$
            $\bar{x}_l = \bar{x}_l^- + K_l(z_l - D_l \bar{x}_l^-)$
        end for
        $\hat{x}_n = \bar{x}_n$
    end for
end
```

7. Object Tracking Test

7.1. Results of Simulation Test

The simulation was performed using the object tracking model and matrices proposed in section IV. Where, the first state is the distance, and the second state is the velocity and the third state is the acceleration, which corresponds to a constant acceleration state model. The moving object model can be described by (1) and (2). For the first simulation test we consider that an object target is disturbed by white Gaussian acceleration noise with a standard deviation of $\sigma_w = 1m/s^2$. The data noise originates from white Gaussian with $\sigma_\xi = 20\,\mathrm{m}$. The simulation of the trajectory was 2000 points with sample time $T = 0.05$ seconds, $P_0 = 0$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $N_opt = 100$.

The RMSE results obtained by the standard and modified filters for CMN, KF and UFIR, from the color factor Ψ 0 to 0.95 are shown in Fig. 1. In the calculation of RMSE with the standard and modified UFIR algorithm for CMN, a Nopt of the same size was used for the entire range of the coloration factor Ψ.

Results of standard UFIR are shown with magenta line, standard KF with black line, and modified UFIR for CMN with red line. For the KF modified for CMN, the object tracking model states were established under ideal conditions, $p = q = 1$, and with errors in the noise statistics, $p \neq 1$, represented with blue line, and $q \neq 1$ with represented with blue dotted line.

Assuming that a complete information about noise is incomplete, in Fig. 1 we showed the relevant filtering errors produced by substituting in the algorithms Q with p^2Q and R with q^2R for $\{p,q\} > 0$ [8], [31]. As can be seen, even slight modifications in error factors ($p = 2, q = 0.5$) make the KF CMN less accurate than the UFIR CMN. It even has a poorer performance than standard filters. It is important to note that the assumption of incomplete noise information did not apply to the standard KF.Since both the standard and modified UFIR filter for CMN do not require knowing the noise conditions Q and R it shows that the UFIR filter is more preferable for object target tracking. To corroborate the effect of a model with incomplete or unknown noise information. We performed another test considering only the highest noise factor $\Psi = 0.95$. For this test, we propose a model with standard deviation of $\sigma_w = 5m/s^2$, The data noise with $\sigma_v = 10m$.The simulated trajectory with 2000 points with a sample time $T = 0.05$ seconds, $P_0 = 0$ on a short horizon $Nopt = 100$.

Derived from the results obtained in the previous test, where the filters modified for CMN showed better results, it was decided to carry out the second test only with the filters modified for CMN. The results obtained are shown in Fig. 2,we showed the relevant filtering errors produced by substituting in the algorithms Q with p^2Q and R with q^2R for $\{p,q\} > 0$, ($p = 2, q = 0.5$).

As can be seen, that before a slight modification of the error factor, a significant change is generated in the performance of the KF CMN algorithm. On the other hand, since UFIR does not require the initial noise conditions, it is more robust to state models in which the information of noise is not available. This gives another proof that the UFIR CMN is more suitable for object tracking process.

7.2. Results of Benchmark Data Test

In this section we work with benchmark data available on [11], the data with which the tests were performed are called ”Car4” and ”SUV”. Taking into account the tests carried out with the simulated data, in the previous subsection, we decided to carry out the state estimation test with the Nopt corresponding to the standard and modified UFIR. The tests

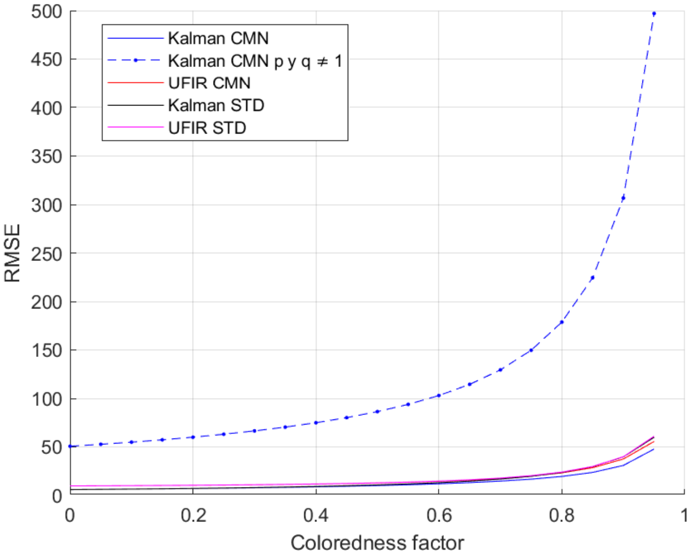

Fig. 1. RMSE results obtained b,y the KF and UFIR standard and modified filters for CMN with Ψ 0 to 0.95.

were performed with the highest noise factor $\Psi = 0.95$ in order to evaluate the robustness of the algorithms.

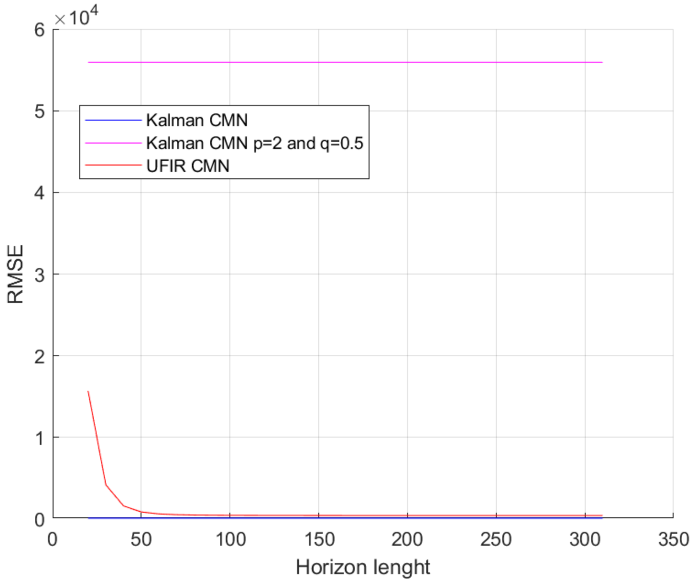

Fig. 2. RMSE results obtained by modified filters for CMN, KF and UFIR ($Nopt = 100$), with the colored factor $\Psi = 0.95$.

For the benchmark data test,"Car4", for the object tracking model we considered that the car target is disturbed by white Gaussian acceleration noise with the standard deviation of $\sigma_w = 3m/s^2$. The data noise (CMN) originates from white Gaussian $\sigma_v = 2m$. The sample time T= 0.05 seconds, $P_0 = 0,$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $N_opt = 110$. The model of a moving target is completed according to what is established in the section IV.

The precision values of each of the filters in the entire intersection over union (IoU) threshold range are shown in Fig.3 and the F-score values in 4. The UFIR CMN and KF CMN filters presented the best performance, with the UFIR CMN being slightly better in the full range of IoU threshold. With this test it can be seen that the standard filters had a poor performance in the estimation of states under colored noise with a high color factor $\Psi = 0.95$. The UFIR CMN and KF CMN filters presented the best performance, where with UFIR CMN beign slightly better than KF CMN in the full range of IoU treshold. Analyzing the F-score metric,the performance is similar to that given by the precision metric. Both metrics serve us to measure the performance of the tracking algorithm. As opposed to precision which focuses on what percentage of the target bounding box was detected; with F-score we can see if there were target losses. In this case, it is observed that the standard Kalman filter presented the lowest value of F-score.

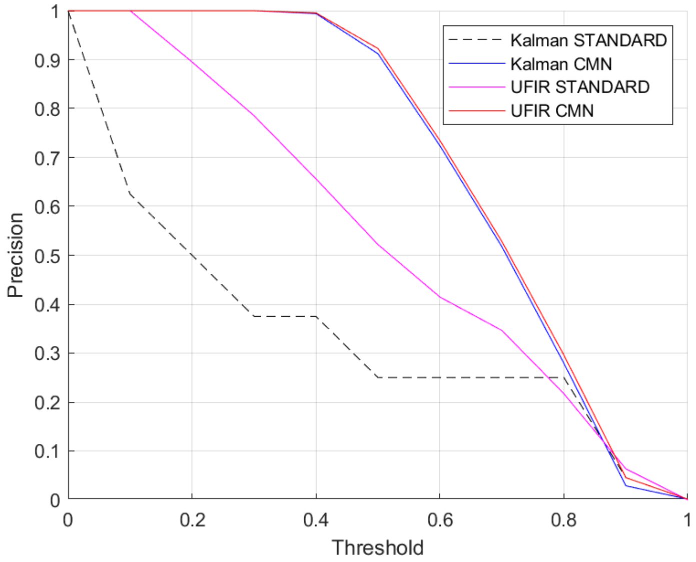

Fig. 3. Precision of benchmark "Car4"($Nopt = 110$),with $\Psi = 0.95$.

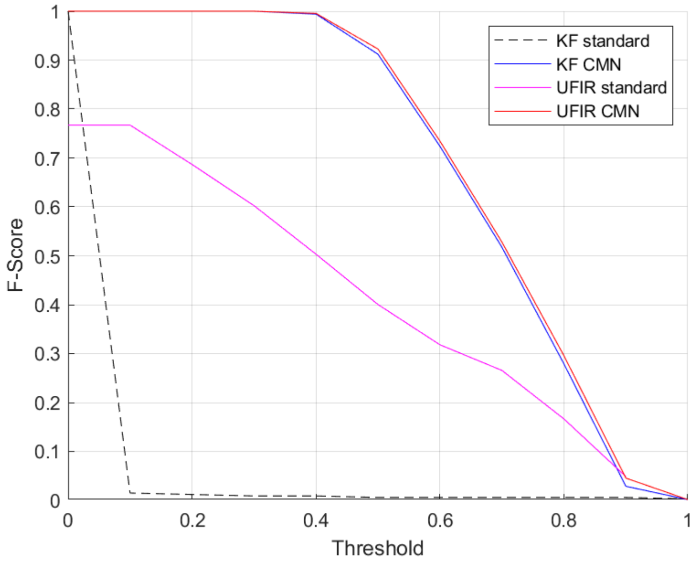

Fig. 4. F-score of benchmark "Car4"($Nopt = 110$),with $\Psi = 0.95$.

For the benchmark data test,"SUV", for the object tracking model we considered that the car target is disturbed by white Gaussian acceleration noise with the standard deviation of $\sigma_w = 10m/s^2$. The data noise (CMN) originates from white Gaussian $\sigma_v = 5m$. The sample time T= 0.05 seconds, $P_0 = 0,$, $Q = \sigma_w^2$, $R = \sigma_V^2$, on a short horizon $N_opt = 70$. The model of a moving target is completed according to what is established in the section IV.

The precision values of each of the filters over the entire Intersection over Union (IoU) threshold range are shown in Fig.5 and the F-score values in 6. In this case, both values are similar, the F-score is slightly lower. The UFIR CMN presented the best performance at a higher value of IoU threshold, being the UFIR CMN slightly better after the threshold of IoU 0.4 for precision and F-score. Overall, the

KF CMN filter was the second best estimator. However, its accuracy and F-score were lower at high IoU values, being outperformed by the standard UFIR filter. This may be because the initial conditions of the motion model are not exactly known. As with the ”Car4” test, it can be seen that the standard Kalman filter had a performance poorly in the estimation of states under colored noise with a high color factor $\Psi = 0.95$.

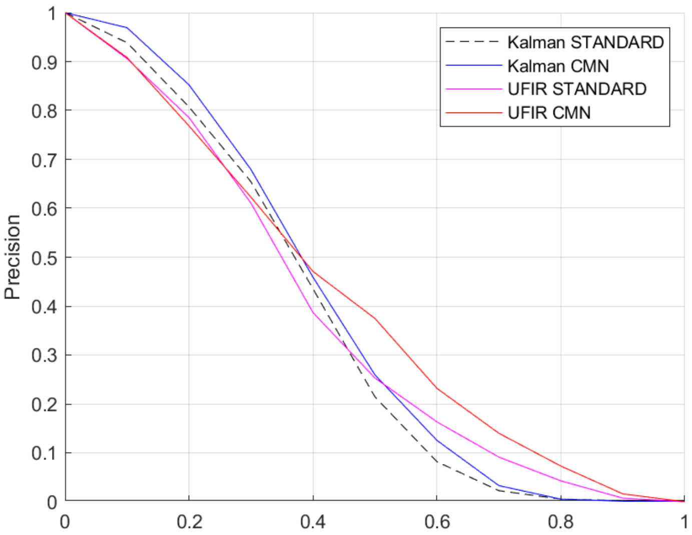

Fig. 5. Precision of benchmark ”SUV”($Nopt = 70$),with $\Psi = 0.95$.

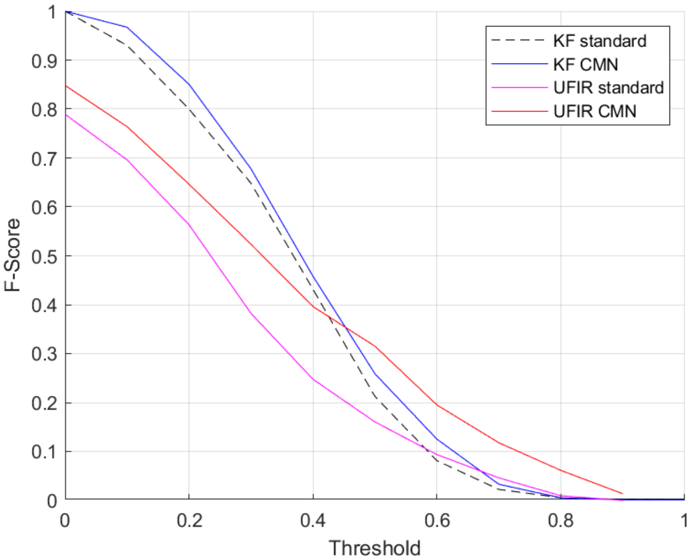

Fig. 6. F-score of benchmark ”SUV”($Nopt = 110$),with $\Psi = 0.95$.

8. Conclusions

Based on the results of this work, we find that, in general, the standard filters, UFIR and KF, presented a low performance in the estimation of states with data in the presence of colored measurement noise and with noise factor Ψ presenting low precision and accuracy values (F-score). The KF CMN filter performs well when used simulated data with ideal conditions and known data and process noise. In the same way, it was shown to be a good estimator with real data under non-ideal conditions, but it is highly dependent on a correct approach to the movement model, as was demonstrated in the estimation with simulated data when using incomplete or unknown noise.

The algorithm UFIR CMN generally obtained better results and proved to be more robust, since it does not require knowing the initial conditions of process noise and data. Its performance was good under non-ideal conditions and with a high noise factor value. These characteristics make it an estimation algorithm with great application in movement models where the information is not well known or the complete information is not available.

Therefore, we conclude that the incorporation of UFIR CMN state estimation algorithms would contribute to the development and improvement of applications and research in the field of object tracking research.

We are currently working on modifications of the UFIR CMN algorithm, since although we consider good results were obtained, we are focused to the improvement in the estimates when the values of Ψ and IoU threshold are the highest. This in order to develop a more robust and efficient algorithm.

References

[1] A. N. Bishop, A. V. Savkin, and P. N. Pathirana, “Vision-based target tracking and surveillance with robust set-valued state estimation,” *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 289–292, 2010.

[2] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *Acm computing surveys (CSUR)*, vol. 38, no. 4, pp. 13–es, 2006.

[3] “Robust visual tracking framework in the presence of blurring by author = Kang, TaeKoo and Mo, YungHak and Pae, DongSung and Ahn, ChoonKi and Lim, MyoTaeg, journal=Measurement, volume=95, pages=50–69, year=2017, publisher=Elsevier.”

[4] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Optimal and unbiased filtering with colored process noise using state differencing,” *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 548–551, 2019.

[5] Y. Yoon, A. Kosaka, and A. C. Kak, “A new kalman-filter-based framework for fast and accurate visual tracking of rigid objects,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1238–1251, 2008.

[6] D. Simon, *Optimal state estimation: Kalman, H_∞, and nonlinear approaches*. Hoboken,NJ: John Wiley & Sons, 2006.

[7] P. Liang, E. Blasch, and H. Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE transactions on image processing*, vol. 24, no. 12, pp. 5630–5644, 2015.

[8] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Kalman and ufir state estimation with coloured measurement noise using backward euler method,” *IET Signal Processing*, vol. 14, no. 2, pp. 64–71, 2020.

[9] Y. S. Shmaliy, “Linear optimal fir estimation of discrete time-invariant state-space models,” *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3086–3096, 2010.

[10] S. Zhao, Y. S. Shmaliy, and C. K. Ahn, “Bias-constrained optimal fusion filtering for decentralized wsn with correlated noise sources,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 727–735, 2018.

[11] (2015) Datasets-visual tracker benchmark. [Online]. Available: http://www.visual-tracking.net

[12] W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge, *Principles of digital image processing*. Springer, 2009, vol. 54.

[13] A. Dave, T. Khurana, P. Tokmakov, C. Schmid, and D. Ramanan, “Tao: A large-scale benchmark for tracking any object,” in *European conference on computer vision*. Springer, 2020, pp. 436–454.

[14] P. Voigtlaender, L. Luo, C. Yuan, Y. Jiang, and B. Leibe, “Reducing the annotation effort for video object segmentation datasets,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3060–3069.

[15] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, “Dvs benchmark datasets for object tracking, action recognition, and object recognition,” *Frontiers in neuroscience*, vol. 10, p. 405, 2016.

[16] O. S. Google. (2020) Open images dataset v6. [Online]. Available: https://storage.googleapis.com/openimages/web/download.html

[17] K. Choeychuen, P. Kumhom, and K. Chamnongthai, “An efficient implementation of the nearest neighbor based visual objects tracking,” in *2006 International Symposium on Intelligent Signal Processing and Communications*. IEEE, 2006, pp. 574–577.

[18] P. Deepak and S. Suresh, “Design and utilization of bounding box in human detection and activity identification,” in *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*. Springer, 2015, pp. 59–70.

[19] M. S. Grewal and A. P. Andrews, *Kalman filtering: Theory and Practice with MATLAB*. Hoboken,NJ: John Wiley & Sons, 2014.

[20] Y. Zhou and S. Suri, “Analysis of a bounding box heuristic for object intersection,” *Journal of the ACM (JACM)*, vol. 46, no. 6, pp. 833–857, 1999.

[21] B. Karasulu and S. Korukoglu, “A software for performance evaluation and comparison of people detection and tracking methods in video processing,” *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 677–723, 2011.

[22] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[23] R. Padilla, W. L. Passos, T. L. Dias, S. L. Netto, and E. A. da Silva, “A comparative analysis of object detection metrics with a companion open-source toolkit,” *Electronics*, vol. 10, no. 3, p. 279, 2021.

[24] E. Ranguelova, B. Weel, D. Roy, M. Kuffer, K. Pfeffer, and M. Lees, “Image based classification of slums, built-up and non-built-up areas in kalyan and bangalore, india,” *European journal of remote sensing*, vol. 52, no. sup1, pp. 40–61, 2019.

[25] F. Sun, H. Li, Z. Liu, X. Li, and Z. Wu, “Arbitrary-angle bounding box based location for object detection in remote sensing image,” *European Journal of Remote Sensing*, vol. 54, no. 1, pp. 102–116, 2021.

[26] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, “Visual tracking: An experimental survey,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[27] C. Xiu, X. Su, and X. Pan, “Improved target tracking algorithm based on camshift,” in *2018 Chinese control and decision conference (CCDC)*. IEEE, 2018, pp. 4449–4454.

[28] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking. part i. dynamic models,” *IEEE Transactions on aerospace and electronic systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[29] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[30] R. G. Brown and P. Y. Hwang, “Introduction to random signals and applied kalman filtering: with matlab exercises and solutions,” *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, 1997.

[31] Y. S. Shmaliy, S. Zhao, and C. K. Ahn, “Unbiased finite impluse response filtering: An iterative alternative to kalman filtering ignoring noise and initial conditions,” *IEEE Control Systems Magazine*, vol. 37, no. 5, pp. 70–89, 2017.

[32] S. Zhao, Y. S. Shmaliy, and F. Liu, “Fast kalman-like optimal unbiased fir filtering with applications,” *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2284–2297, 2016.

[33] D. K. Ryu, C. J. Lee, S. K. Park, and M. T. Lim, “Unbiased fir filtering with incomplete measurement information,” *International Journal of Control, Automation and Systems*, vol. 18, no. 2, pp. 330–338, 2020.