

Research Article

Alternating Projections Filtering Algorithm to Track Moving Objects

Youssef Qranfal 

Department of Applied Mathematics, Wentworth Institute of Technology, Boston, MA 02115, USA

Correspondence should be addressed to Youssef Qranfal; qranfaly@wit.edu

Received 8 November 2018; Revised 24 March 2019; Accepted 23 April 2019; Published 13 May 2019

Academic Editor: Oluwale D. Makinde

Copyright © 2019 Youssef Qranfal. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

An interest is often present in knowing evolving variables that are not directly observable; this is the case in aerospace, engineering control, medical imaging, or data assimilation. What is at hand, though, are time-varying measured data, a model connecting them to variables of interest, and a model of how to evolve the variables over time. However, both models are only approximation and the observed data are tainted with noise. This is an ill-posed inverse problem. Methods, such as Kalman filter (KF), have been devised to extract the time-varying quantities of interest. These methods applied to this inverse problem, nonetheless, are slow, computation wise, since they require large matrices multiplications and even matrix inversion. Furthermore, these methods are not usually suitable to impose some constraints. This article introduces a new iterative filtering algorithm based on alternating projections. Experiments were run with simulated moving projectiles and were compared with results using KF. The new optimization algorithm proves to be slightly more accurate than KF, but, more to the point, it is much faster in terms of CPU time.

1. Introduction

Tracking moving objects or a substance is a common optimization and control problem [1] in various fields such as aerospace [2, 3], medical imaging [4–6], data assimilation [7–9], and video surveillance [10–12]. Methods, such as Kalman filter, have been devised to solve this problem. However, these methods come with a price, for large systems, of being slow in computation since they involve matrix-matrix multiplication and even matrix inversion. Remedies, such as Expectation Maximization Filter (EMF) algorithm, have been proposed including when nonnegative results are desired [13–16]. A new remedy is yet introduced here.

The novel filtering algorithm presented in this article aims, as does KF, to find an estimate \hat{x}_k , $k = 1, \dots, K$, to the unknown x_k of the problem modeled by the two linear space-state equations,

$$\begin{aligned} x_k &= A_k x_{k-1} + \mu_k, \\ z_k &= H_k x_k + v_k. \end{aligned} \quad (1)$$

The error vector μ_k has $\mathbb{E}(\mu_k) = 0$ and $\mathbb{E}(\mu_k \mu_k^T) = Q_k$; the latter is the covariance of the error in modeling the transition

from x_{k-1} to x_k . $\mathbb{E}(v_k) = 0$ and $\mathbb{E}(v_k v_k^T) = R_k$ are the mean and covariance, respectively, of the noise vector v_k . The assumption is that the error and noise are white so that

$$\begin{aligned} Q_k &= \sigma_k^2 I, \\ R_k &= \text{diag}(\gamma_k), \end{aligned} \quad (2)$$

where I is the identity matrix with order known from context; $\text{diag}(a)$ denotes the square matrix that has the $a_i > 0$ in its main diagonal and 0 otherwise. Both matrices Q_k and R_k are assumed to be diagonal. The new algorithm is then numerically tested to solve a reconstruction problem arising in object tracking.

Tracking a moving object (or moving objects) with large unknown variables and few given data, as in aerospace, medical imaging, or data assimilation, is an ill-posed inverse problem. This ill-posedness is further inflated by physical degradation of the acquired data causing noise. Filtering algorithms, as the one presented in this article, are therefore very suitable tools.

The remainder of the paper is organized as follows. Section 2 describes the stochastic modeling of the state

evolution and projection in space. The state evolution and space linear models are the basis to apply the new method. Then Section 3 reviews KF and introduces the new filtering algorithm. The application of the new filter to tracking moving objects is covered in Section 5. Section 6 details the simulations and numerical experiments of moving tracking objects using both KF and the new filter; these corroborate the effectiveness of this new algorithm in terms of convergence and CPU time. Section 7 closes the paper by summarizing the findings.

2. Problem Formulation

2.1. A Stochastic Model of Moving Objects. A new filtering algorithm is introduced. Without loss of generality, it is illustrated, with an example of tracking moving objects. It could be applied to any instance where KF was and is used when modeled by the two linear state-space equations (3) and (9). For instance, the number of photons in [4, 6] is the unknown x_k of both these equations.

To illustrate the new algorithm, tracking moving objects is studied here where detectors, such as radars or cameras, measure their positions over time. The goal is to estimate their positions, velocities, and/or acceleration, all at once, given a few noisy measurements of their positions. Kalman filter can average out the noise of the measured data and gives somehow smooth tracks of these over time moving objects in space [11, 17]. Filtering is a widely used method in engineering where a filtering algorithm, such as KF, reduces the noise from signals while keeping the utile information. As in the case of KF, both a process/evolution model and a measurement equation are used to estimate the unknowns. Note that the unknown vectors x_k of both state-space equations (3) and (9) in this illustrating example include the positions, velocities, and acceleration of the moving objects.

Velocity and acceleration can be easily derived from the position. Someone could be interested in finding only the positions and can easily derive the velocities and acceleration therefrom; however, having velocities and acceleration helps to build the state evolution of (3). In addition, having more than the positions in the unknown x_k adds more to their components, therefore increasing the size of the problem to illustrate the algorithm.

Detectors are exploited to register the positions of the moving objects. The problem is then modeled as follows. Let $t_k, k = 1, \dots, K$ be an index of a sequence of acquisition times, J the total number of unknowns, and I the total number of detectors during the time t_k . The variables $x_k \in \mathbb{R}^J$ and $z_k \in \mathbb{R}^I$ denote the state vector containing the unknowns of the moving objects and the measured data during the $t_k = (\Delta t)_k$ time, respectively. The unknowns include the 2D or 3D components of positions, velocities, and/or acceleration. The j^{th} entry in the vector x_k has a component of either the position, velocity, and/or acceleration during the time t_k . The observations z_1, z_2, \dots, z_K are random vectors. The i^{th} entry in the vector z_k measures the component of position of a moving object registered at the i^{th} detector during the time t_k . Furthermore, each observation z_k depends on x_k only.

The sequence x_1, x_2, \dots, x_K satisfies Markov property with unknown time-varying transition matrix $A_k \in \mathbb{R}^{J \times J}$. That is to say, the present state x_k depends only on the previous one x_{k-1} according to the following linear equation:

$$x_k = A_k x_{k-1} + \mu_k, \quad (3)$$

where μ_k is the error random vector in modeling the transition from x_{k-1} to x_k with $\mathbb{E}(\mu_k)$ equal to zero and covariance matrix $\mathbb{E}(\mu_k \mu_k^T) = Q_k$.

2.2. Example. To make sense of this linear evolution model, consider the following basic 2D example with a constant time increment $(\Delta t)_k = \Delta t$. Assume that the state of a moving object is the vector

$$x_k = [s_{k,x} \ s_{k,y} \ v_{k,x} \ v_{k,y} \ a_{k,x} \ a_{k,y}]^T, \quad (4)$$

where $s_{k,x}$ and $s_{k,y}$ are the horizontal and vertical components of position, respectively, with corresponding velocity components $v_{k,x}$ and $v_{k,y}$ and acceleration components $a_{k,x}$ and $a_{k,y}$. That is, $s_k = [s_{k,x} \ s_{k,y}]^T$, $v_k = [v_{k,x} \ v_{k,y}]^T$, and $a_k = [a_{k,x} \ a_{k,y}]^T$. The basic kinematic equations with constant acceleration are

$$s_k = s_{k-1} + v_{k-1} \Delta t + 0.5 a_{k-1} \Delta t^2, \quad (5)$$

$$v_k = v_{k-1} + a_{k-1} \Delta t, \quad (6)$$

$$a_k = a_{k-1}. \quad (7)$$

Equations (5)-(7) can be written in a matrix form,

$$\begin{bmatrix} s_{k,x} \\ s_{k,y} \\ v_{k,x} \\ v_{k,y} \\ a_{k,x} \\ a_{k,y} \end{bmatrix}_{x_k} = \underbrace{\begin{bmatrix} 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_k} \underbrace{\begin{bmatrix} s_{k-1,x} \\ s_{k-1,y} \\ v_{k-1,x} \\ v_{k-1,y} \\ a_{k-1,x} \\ a_{k-1,y} \end{bmatrix}}_{x_{k-1}}. \quad (8)$$

2.3. Assumptions. Let $(h_k)_{ij}$ be the quantity that links the unknown in position j during the acquisition time t_k to detector i . The time-varying matrix defined by $H_k = [(h_k)_{ij}]$ is called projection or observation matrix. This matrix H_k accounts for the relationship between the measurement vector z_k and the state vector x_k . The observations/measurements and state vectors are related by the linear equation

$$z_k = H_k x_k + \nu_k, \quad (9)$$

where $\mathbb{E}(\nu_k) = 0$ and $\mathbb{E}(\nu_k \nu_k^T) = R_k$ are the mean and covariance, respectively, of the noise vector ν_k .

For instance, consider example 2.2 where the state vector x_k is of size 6. Two noisy measurements, $z_k = [z_{k,x}, z_{k,y}]^T$,

namely, the horizontal and vertical components of the position, are known. Then the 2×6 matrix H_k is

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (10)$$

Note that this is just an illustrating example with very small sizes; x_k , H_k , and A_k have sizes 6×1 , 2×6 , and 6×6 , respectively. In practice, the sizes are of hundreds, thousands, or even more.

The optimal \hat{x}_k is to be found given the data z_k , the evolution matrix A_k , and the observation system matrix H_k at each time step k . Equations (3) and (9) are the state-space form of a particular case of a more general filtering problem [18, 19]. The actual model is a linear dynamic system for which the analytic filtering solution is given by the KF [20].

The covariance matrices Q_k and R_k being diagonal and the error as well as the noise being white are three assumptions adopted in this paper. In case one assumption or more are not satisfied, a prewhitening process [13] can be used to bring a general setting to the one of this paper.

3. Filtering and Reconstruction

The index k is dropped whenever it is needed for convenience. For instance, at each instance time k , the state vector x_k , the data vector z_k , and the matrix H_k are referred to as just x , z , and H , respectively. The moving objects problem amounts to finding $x \in \mathbb{R}^I$ solution of the linear equation

$$z = Hx + v, \quad (11)$$

where $z \in \mathbb{R}^I$, $H \in \mathbb{R}^{I \times J}$ are the given observation data vector and the observation system matrix, respectively. The vector $v \in \mathbb{R}^I$ represents the additive noise in recording the data z . A new method, based on alternating projections over convex sets and compared to KF, is introduced, but, first, a review of the KF follows.

3.1. Kalman Filter. The KF update [13, 20] is the following: Given an unbiased estimate \hat{x}_{k-1} of the state vector x_{k-1} , the prior estimate of x_k based solely on the activity dynamics is

$$y_k = A_k \hat{x}_{k-1}. \quad (12)$$

The estimate \hat{x}_k will have the form (refer, for instance, to [6])

$$P_k = A_k P_{k-1} A_k^\top + Q_k, \quad (13)$$

$$K_k = P_k H_k^\top (H_k P_k H_k^\top + R_k)^{-1}, \quad (14)$$

$$\hat{x}_k = y_k + K_k (z_k - H_k y_k). \quad (15)$$

P_k and P_{k-1} in (13) are the covariances of the estimated activity \hat{x} at time k and $k-1$, respectively. As mentioned before, the matrices' sizes involved in (13) and (14) are of the order of hundreds, thousands, or even more. Therefore, a major drawback of KF is the matrix-matrix multiplications involved in (13) and (14) and matrix inversion involved in (14).

Attempts have been made to rectify these two shortcomings; see, for instance, [18, 19] for more details. The goal of the new filter is manifold, but without imposing nonnegativity as in [13, 14]. It is intended to find a substitute algorithm to KF that filters out errors from modeling the dynamical system and the noise from the data. It will insure temporal regularization and will be an optimal recursive estimate. In addition, it does not require the storage of past measurement data, use matrix-matrix multiplications, or necessitate any matrix inversion. Furthermore, it does not need to calculate, update, or store any covariance matrix. The aim is then to keep properties of KF while ameliorating it by requesting few more. Each recursive step in the new approach is an iterative operation that involves only matrix-vector multiplication. This should then handle the problems of huge number of variables, such as the case in aerospace.

4. Alternating Projections Minimization

The approach here to solving for both linear state-space equations (3) and (9) is by solving first, in this section, a regularization problem (16) that involves a regularization parameter α . Once the scheme in solving problem (16) is developed, both (3) and (9) are put together, at each time step k , in one regularization problem (30) similar to problem (16) using a regularization parameter α_k updated at each time step k . As for the variable y of problem (16), it is substituted by the state transition in time $A_k x_{k-1}$ vector of (3) and (12). This process of solving (3) and (9) using the scheme developed in this section is done in the Section 5. Thus, a regularization problem is introduced and solved next.

The authors in [13, 14, 21] studied a regularization problem. This paper considers the following similar problem, for $0 \leq \alpha \leq 1$, using the Euclidian distance instead:

$$\min F(x) = \alpha \|z - Hx\|^2 + (1 - \alpha) \|y - x\|^2. \quad (16)$$

The new filtering algorithm takes care of both underdetermined and overdetermined cases. Moreover, an iterative method, within the framework of alternating projections, is developed.

First, two convex sets are defined of IJ -dimensional vectors similar to what is done in [21–24] as

$$\mathcal{R} = \left\{ r = \{r_{ij}\} \in \mathbb{R}^{IJ} \mid \text{for all } i, z_i = \sum_{j=1}^J r_{ij} \right\}, \quad (17)$$

$$\mathcal{Q} = \{q = q(x) = \{q_{ij} = H_{ij} x_j\} \text{ for some } x\}.$$

The sets \mathcal{R} and \mathcal{Q} are nonempty, closed, and convex in \mathbb{R}^{IJ} . Recall the function F in (16); $d(r, q)$ is also defined as

$$d(r, q) = F(x) = \alpha \|q - r\|^2 + (1 - \alpha) \|y - x\|^2. \quad (18)$$

Expand the most right hand side of expression (18).

$$d(r, q) = \alpha \sum_{i=1}^I \sum_{j=1}^J (H_{ij}x_j - r_{ij})^2 + (1 - \alpha) \sum_{j=1}^J (y_j - x_j)^2 \quad (19)$$

The alternating projections minimization method works as follows. First start with an initial guess \hat{x}^0 . Then, having obtained the ℓ^{th} iterate, set $q^\ell = q(x^\ell)$. Afterwards perform the following two steps:

Step 1. Minimize $d(r, q^\ell)$ with respect to r to get $r^{\ell+1}$, which is in the convex set \mathcal{R} .

Step 2. Minimize $d(r^{\ell+1}, q(x))$ with respect to x to get $x^{\ell+1}$. The corresponding $q(x^{\ell+1})$ is in the convex set \mathcal{Q} .

The sequence $\{x^\ell\}$ converges to a limit x^∞ , which is the estimate \hat{x} we are looking for. This iterative scheme is applied to the problem at hand while performing step 1 using (19):

$$\frac{\partial d}{\partial r_{ij}} = 0 \implies r_{ij} = H_{ij}x_j + c_i, \quad \text{for some } c \in \mathbb{R}^I. \quad (20)$$

However $r_{ij} \in \mathcal{R}$ so that $\sum_{j=1}^J r_{ij} = z_i$; therefore $z_i = (Hx)_i + Jc_i$, where $(Hx)_i = \sum_{j=1}^J H_{ij}x_j$, which leads to

$$c_i = \frac{1}{J} (z_i - (Hx)_i), \quad (21)$$

$$r_{ij} = H_{ij}x_j + \frac{1}{J} (z_i - (Hx)_i).$$

Starting with a guess \hat{x}^0 , the update expression is

$$r_{ij} = r_{ij}^{\ell+1} = r(x_j^\ell) = H_{ij}x_j^\ell + \frac{1}{J} (z_i - (Hx^\ell)_i). \quad (22)$$

Now, Step 2 is performed using (19):

$$\frac{\partial d}{\partial x_j} = 0 \implies \alpha \sum_{i=1}^I H_{ij} (r_{ij} - H_{ij}x_j) + (1 - \alpha) (y_j - x_j) = 0. \quad (23)$$

Therefore,

$$\alpha \left(\sum_{i=1}^I H_{ij} r_{ij} - \sum_{i=1}^I H_{ij}^2 x_j \right) + (1 - \alpha) y_j - (1 - \alpha) x_j = 0. \quad (24)$$

So that,

$$\alpha \sum_{i=1}^I H_{ij} r_{ij} + (1 - \alpha) y_j = \alpha \sum_{i=1}^I H_{ij}^2 x_j + (1 - \alpha) x_j, \quad (25)$$

$$\alpha \sum_{i=1}^I H_{ij} r_{ij} + (1 - \alpha) y_j = x_j \left(\alpha \sum_{i=1}^I H_{ij}^2 + (1 - \alpha) \right).$$

This leads to the update formula,

$$x_j^{\ell+1} = \frac{\alpha \sum_{i=1}^I H_{ij} r_{ij}^{\ell+1} + (1 - \alpha) y_j}{\alpha \sum_{i=1}^I H_{ij}^2 + (1 - \alpha)}. \quad (26)$$

Both update formulas (22) and (26) could be combined in only one:

$$x_j^{\ell+1} = \frac{\alpha \sum_{i=1}^I (H_{ij}^2 x_j^\ell + N H_{ij} (z_i - (Hx^\ell)_i)) + (1 - \alpha) y_j}{\alpha \sum_{i=1}^I H_{ij}^2 + (1 - \alpha)}, \quad (27)$$

where $N = 1/J$. Starting with a guess \hat{x}^0 , the sequence $\{x^\ell\}$ converges to a limit x^∞ , which is the desired estimate \hat{x} . Let us apply the above update formula (27) to the tracking moving objects problem.

5. Application to Moving Objects

The goal here is to solve for both linear state-space equations (3) and (9) by using the finding of Section 4. It goes as follows. The error and noise covariance matrices in tracking the moving objects problem are modeled as diagonal matrices with nonnegative entries; refer to Section 2.1. That is,

$$Q_k = \sigma_k^2 I, \quad (28)$$

$$R_k = \gamma_k^2 I, \quad (29)$$

where I is the identity matrix; its dimension is known from context and $\sigma_k \geq 1$. In case $0 < \sigma_k < 1$, it suffices, for instance, to multiply matrix Q_k by σ_k^4 and use $1/\sigma_k^2$ instead of σ_k^2 . In case when $R_k = \text{diag}(a)$ in expression (29), $R_k^{-1/2} = \text{diag}(1/\sqrt{a})$ is used instead of $(1/\gamma_k)I$ ($\gamma_k > 0$).

Recall that the aim is to find an estimate \hat{x}_k , $k = 1, \dots, K$, to the unknown x_k of the problem modeled by the two linear space-state equations (3) and (9).

The functional to be minimized at each recursion step k is

$$F(x_k) = \|z_k - H_k x_k\|_{R_k^{-1}}^2 + \|y_k - x_k\|_{Q_k^{-1}}^2$$

$$= \left\| \frac{1}{\gamma_k} z_k - \frac{1}{\gamma_k} H_k x_k \right\|^2 + \left\| \frac{1}{\sigma_k} y_k - \frac{1}{\sigma_k} x_k \right\|^2$$

$$= \left\| \frac{1}{\gamma_k} z_k - \frac{1}{\gamma_k} H_k x_k \right\|^2 + \frac{1}{\sigma_k^2} \|y_k - x_k\|^2$$

$$\begin{aligned}
&= \frac{\sigma_k^2 - 1}{\sigma_k^2} \left\| \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} z_k - \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} H_k x_k \right\|^2 \\
&\quad + \frac{1}{\sigma_k^2} \|y_k - x_k\|^2 \\
F(x_k) &= \alpha_k \left\| \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} z_k - \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} H_k x_k \right\|^2 \\
&\quad + (1 - \alpha_k) \|y_k - x_k\|^2,
\end{aligned} \tag{30}$$

where $\alpha_k = (\sigma_k^2 - 1)/\sigma_k^2$. Note that $0 \leq \alpha_k \leq 1$ since $\sigma_k \geq 1$, which makes $F(x_k)$ the same functional to be minimized as in (16).

It suffices now to use the change of variables, given in step 2 of the Alternating Projections Filter (APF) algorithm 5.1 that follows, and to recall that the predicted state vector y_k is $A_k x_{k-1}$. The iterative method, based on alternating projections and given by the updated formulas (22) and (26), or the combined formula (27) at each time step k , is used. The clustering point x^∞ will be the estimate state \hat{x}_k that is solved for using (3) and (9). Thus the following APF algorithm is obtained.

5.1. Alternating Projections Filter Algorithm

- (1) To solve the linear state-space problem given by (3) and (9), start with a \hat{x}_0 and set $N = 1/J$. For $k = 1, \dots, K$ execute the following steps.
- (2) Assume the recursive steps up to time $k - 1$ are done; carry the change of variables

$$\begin{aligned}
\alpha &= \frac{\sigma_k^2 - 1}{\sigma_k^2}, \\
P &= \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} H_k = \frac{1}{\gamma_k \sqrt{\alpha}} H_k, \\
b &= \frac{\sigma_k}{\gamma_k \sqrt{\sigma_k^2 - 1}} z_k = \frac{1}{\gamma_k \sqrt{\alpha}} z_k.
\end{aligned} \tag{31}$$

- (3) To get \hat{x}_k , start with $s^0 = \hat{x}_{k-1}$.
- (4) Make $c = y_k = A_k \hat{x}_{k-1}$.
- (5) Do $r_{ij}^{\ell+1} = P_{ij} s_j^\ell + N(b_i - (Ps^\ell)_i)$.
- (6) Compute

$$s_j^{\ell+1} = \frac{\alpha \sum_{i=1}^I P_{ij} r_{ij}^{\ell+1} + (1 - \alpha) c_j}{\alpha \sum_{i=1}^I P_{ij}^2 + (1 - \alpha)}, \quad \ell = 0, 1, \dots \tag{32}$$

- (7) The update formula for the next estimate is $\hat{x}_k = s^\infty$, where s^∞ is the cluster point of the sequence $(s^\ell)_{\ell \in \mathbb{N}}$.

Observe that the two steps (5) and (6) could be combined into one step as was done before in (27).

$$\begin{aligned}
s_j^{\ell+1} &= \frac{\alpha \sum_{i=1}^I (P_{ij} s_j^\ell + N P_{ij} (b_i - (Ps^\ell)_i)) + (1 - \alpha) c_j}{\alpha \sum_{i=1}^I P_{ij}^2 + (1 - \alpha)} \\
&= \frac{\alpha (s_j^\ell \sum_{i=1}^I P_{ij}^2 + N \sum_{i=1}^I P_{ij} (b_i - (Ps^\ell)_i)) + (1 - \alpha) c_j}{\alpha \sum_{i=1}^I P_{ij}^2 + (1 - \alpha)} \tag{33} \\
&\quad \ell = 0, 1, \dots
\end{aligned}$$

Note how the iterate $s_j^{\ell+1}$ in (33) is formed as a ratio of two convex combinations. The numerator is a convex combination of the predicted $(y_k)_j$, associated with the same coefficient $1 - \alpha$ as in problem (16), relying only on the evolution model, and of the calculated APF iterate associated with the same coefficient α as in problem (16) as well, relying only on the observation model. The denominator is also a convex combination; it is a weighting factor for the updated vector estimate. The APF does not involve any matrix-matrix multiplication or any matrix inversion. It is iterative, involves only matrix-vector multiplication, and does not need to calculate, update, or store any covariance matrix.

The temporal regularization parameter α is well defined and takes values between 0 and 1 when σ_k varies between 1 and ∞ . If $\alpha = 0$ at each time step k , so that $\sigma_k = 1$, then \hat{x}_k is $A_k \hat{x}_{k-1}$ in (33). That is, the observations are discarded completely to the extent that only the evolution model is relied on. This defeats the purpose of using the full power of detectors. When $\alpha = 1$ at each time step k , the predicted y_k in step (4) is not needed in step (6). Indeed using (33),

$$\begin{aligned}
s_j^{\ell+1} &= \frac{\alpha (s_j^\ell \sum_{i=1}^I P_{ij}^2 + N \sum_{i=1}^I P_{ij} (b_i - (Ps^\ell)_i)) + (1 - \alpha) c_j}{\alpha \sum_{i=1}^I P_{ij}^2 + (1 - \alpha)}, \\
&= \frac{s_j^\ell \sum_{i=1}^I P_{ij}^2 + N \sum_{i=1}^I P_{ij} (b_i - (Ps^\ell)_i)}{\sum_{i=1}^I P_{ij}^2}, \\
s_j^{\ell+1} &= s_j^\ell + \frac{\sum_{i=1}^I P_{ij} (b_i - (Ps^\ell)_i)}{J \sum_{i=1}^I P_{ij}^2}.
\end{aligned} \tag{34}$$

Thus, with the above update (34), the Landweber iteration [25] is retrieved in the static case. Surely, choosing $\alpha = 1$ means that $(\sigma_k - 1)/\sigma_k = 1$ or simply $\sigma_k = \infty$. That is, the covariance matrix in the transition equation (3) is very huge, which implies there is no confidence at all in the evolution model. In other words, the evolutionary state of the variable is discarded. Only the observations z_k are meaningful in finding the $\hat{x}_k = \hat{x}$, which is then a stationary state as it should be.

6. Numerical Experiment

Data Availability. The simulated data used to support the findings of this study are described as follows. The APF algorithm 5.1, presented here in the linear case, is validated

with moving objects while modifying the setting of example 2.2 to have a fairly large size problem. To achieve this goal, the APF algorithm is applied to an example where there are $(J - 1)$ different moving objects starting at the same position with the same velocity that varies over time.

The experiment is done with different large values of J , the number of unknowns in the state vector x_k , to compare results to KF for accuracy and efficiency. Indeed, the goals of this comparison are twofold. First it makes sure that the APF algorithm gives similar results to those of KF; this serves to compare the quality of the outputs of both algorithms for accuracy. Second it compares the CPU running times of both APF and KF for efficiency. The KF algorithm is used here in its conventional/classical formulation. Furthermore, matrices involved in this illustrative example are quite sparse. Both algorithms APF and KF, as they are used here, do not use this sparsity property to decrease the running times. Thus, this approach shows and compares the capacity of both algorithms in a more general case, including when matrices might not be sparse.

The $t_k = (\Delta t)_k = \Delta t$ is set and the test is done with different values of Δt . The I simulated camera detectors measure only the position with some error but not the velocity, so that $I = J - 1$. The evolution matrices are taken as $A_k = A$ of size $J \times J$. This matrix is

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 & \Delta t \\ 0 & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & 0 & \vdots \\ 0 & \cdots & 0 & 1 & \Delta t \\ -1 & 0 & \cdots & 0 & 1 \end{bmatrix}. \quad (35)$$

The value -1 in the $(J, 1)$ position in the matrix A is arbitrary; its sole purpose is to make sure that the velocity is changing over time. The unknown $J \times 1$ state vector is represented as

$$x_k = [s_{k,1} \ \cdots \ s_{k,J-1} \ v_k]^\top, \quad (36)$$

and the projection/measurements matrices $H_k = H$, where H is of size $I \times J$, are

$$H = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \cdots & 0 & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \quad (37)$$

The experiment is run with an initial state vector of size J where $(x_0)_1 = \cdots = (x_0)_{J-1} = 3$ m and $(x_0)_J = v_0 = 5$ m/s²; both values 3 and 5 are arbitrary, so that

$$x_0 = [3 \ \cdots \ 3 \ 5]^\top. \quad (38)$$

The measured data is simulated with the Gaussian noise process in (39). It counts for constant and arbitrary acceleration,

TABLE 1: Experiment parameters.

parameter	value
x_0	$[3 \ \cdots \ 3 \ 5]^\top$
$(\Delta t)_k$	0.05
accel_k	0.6
DataNoise_k	10
α_k	0.7
σ_k^2	10/3
γ_k^2	100
$Q_k = \text{accel}_k^2 \sigma_k^2 I$	1.2 I
J	500

$\text{accel} = 0.6$ m/s², to be added to each position measurement, so that

$$\text{ProcessNoise} = \text{accel} \cdot \frac{\Delta t^2}{2} \cdot \text{randn1}, \quad (39)$$

where randn1 is a $J \times 1$ random vector of the standard normal distribution. The current velocity is updated from the previous one $v_{k-1} = (x_{k-1})_J$ as

$$(x_k)_J = v_k = (x_{k-1})_J + \text{accel} \cdot \Delta t. \quad (40)$$

The current x_k is then updated according to

$$x_k = Ax_{k-1} + \text{ProcessNoise}. \quad (41)$$

The state vector x_k in (41) is updated with ProcessNoise as defined in (39) and also with ProcessNoise equal to zero; no significant change has been noticed. The data or measurements z_k at time t_k are simulated with the state vector, without the last component which contains the velocity, with an added Gaussian noise according to

$$z_k = Hx_k + \text{DataNoise} \cdot \text{randn2}, \quad (42)$$

where DataNoise is some arbitrary value and randn2 is a $J \times 1$ random vector of the standard normal distribution. As for KF, the covariance matrices Q_k are set to be equal to Q with $Q = \sigma^2 I$ and $Q = \text{accel}^2 \sigma^2 I$; a slight improvement for KF in terms of accuracy is noticed. Recall that we set $\alpha = (\sigma^2 - 1)/\sigma^2$; choosing this regularization parameter depends on how much trust is put in each of both state-space equations. The example presented in Table 1 has $\alpha = 0.7$. The results presented in Figure 1 over 100 s are done using parameters summarized in Table 1.

The three curves in Figure 1 are averages of their corresponding $J - 1$ positions. The latter represent the simulated moving objects, reconstructed positions using KF, and reconstructed positions using the APF algorithm 5.1. Taking averages of the positions is for plotting and comparing errors purposes only. Averages are not taken over the time interval Δt , but are rather averages of all the $J - 1$ positions at every

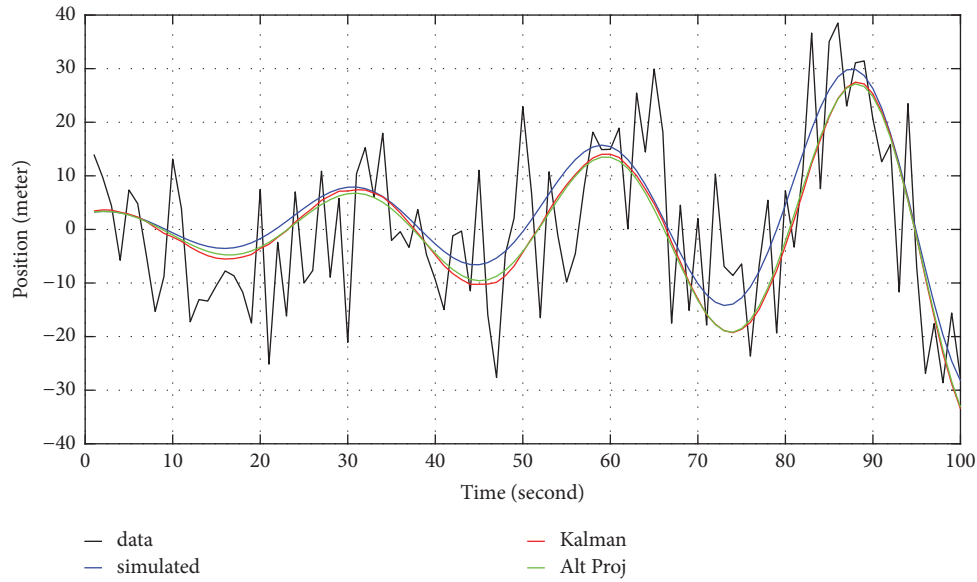


FIGURE 1: Tracking of moving objects: simulated positions in blue, noisy measurements of positions in black, Kalman filter (KF) recovering of positions in red, and Alternating Projections Filter (APF) recovering of positions in green. Running CPU times of both algorithms depend on the size J and the size of time step Δt . As J increases and/or Δt decreases, time of both APF and KF increases; however APF is much faster than KF. Indeed, APF is over 25 times faster than KF with size $J = 2000$ of the unknown state vector and time step $\Delta t = 0.05$ s over 100 s.

time step k . The sampling interval has no impact on the resulting averages.

The relative error between the simulated and both reconstructed curves has a median around 30%. Running CPU times of both algorithms depend on the size J and the size of time step Δt . As J increases and/or Δt decreases, time of both APF and KF increases; however APF is much faster than KF. Indeed, APF is over 25 times faster than KF with size $J = 2000$ of the unknown state vector and time step $\Delta t = 0.05$ s, corresponding to $K = 100$, over 100 s.

7. Conclusion

A novel filtering algorithm APF, to be applied to the linear case, was presented; it could be implemented in aerospace and other fields. It could also be applied, for instance, anytime tracking of moving objects is desired, such as in medical imaging and data assimilation where large size systems are involved. Experiments were run comparing APF and KF in terms of accuracy and computer speed. Quality of reconstructed curves is about the same in both algorithms, although APF performs slightly better than KF. More importantly, APF is up to 25 times faster than KF, seconds instead of minutes. As it is the case with KF, APF algorithm filters out errors from modeling the dynamical system and the noise from the data. Both algorithms insure temporal regularization and output an optimal recursive estimate. However, APF does not use any matrix-matrix multiplication and does not necessitate any matrix inversion. Furthermore, APF does not need to calculate, update, or store any covariance matrix. This is not the case for KF regarding these last three properties. Indeed, these three properties are at the heart of making APF take much less CPU time

compared to KF, so that APF is very suitable for large scale systems such as the ones in aerospace. APF could be used in any discipline which has used, for instance, KF or in any field that is interested in time-varying variables such as financial risk assessment/evaluation and forecasting or control. The results substantiate the efficiency of this novel APF algorithm.

Nomenclature

k :	Time parameter
K :	Maximum value of k
x_k :	Position vector of a moving object at time k
A_k :	Evolution matrix from position x_{k-1} to position x_k
z_k :	Measured data vector at time k
H_k :	System matrix relating x_k to z_k
$\mathbb{E}(w)$:	Expected value of a vector w
$\mathbb{E}(ww^T)$:	Covariance matrix of a vector w
μ_k :	Error vector in modeling the transition from x_{k-1} to x_k
Q_k :	Covariance of μ_k
v_k :	Noise vector of measured z_k
R_k :	Covariance of v_k .

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares no conflicts of interest.

References

- [1] J. Ding, G. Zhang, and L. Zhao, "Urban and indoor weak signal tracking using an array tracker with MVA and nonlinear filtering," *Journal of Applied Mathematics*, vol. 2014, Article ID 107156, 10 pages, 2014.
- [2] S. F. Schmidt, "The Kalman filter—its recognition and development for aerospace applications," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 4–7, 1981.
- [3] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, 2010.
- [4] M. Kervinen, M. Vauhkonen, J. P. Kaipio, and P. A. Karjalainen, "Time-varying reconstruction in single photon emission computed tomography," *International Journal of Imaging Systems and Technology*, vol. 14, no. 5, pp. 186–197, 2004.
- [5] J. Qranfal and G. Tanoh, "Regularized Kalman filtering for dynamic SPECT," *Journal of Physics: Conference Series*, vol. 124, Article ID 012042, 2008.
- [6] J. Qranfal, "Incorporating nonnegativity and spatial regularization constraints using proximal projections," *International Journal of Pure and Applied Mathematics*, vol. 81, no. 2, pp. 267–304, 2012.
- [7] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble kalman filter technique," *American Meteorological Society*, vol. 126, pp. 796–811, 1998.
- [8] P. L. Houtekamer, H. L. Mitchell, G. Pellerin et al., "Atmospheric data assimilation with an ensemble Kalman filter: results with real observations," *Monthly Weather Review*, vol. 133, no. 3, pp. 604–620, 2005.
- [9] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer-Verlag, Berlin, Germany, 2nd edition, 2009.
- [10] A. Czyzewski and P. Dalka, "Examining kalman filters applied to tracking objects in motion," in *Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 175–178, Klagenfurt, Austria, May 2008.
- [11] C. Suliman, C. Cruceru, and F. Moldoveanu, "Kalman filter based tracking in an video surveillance system," *Advances in Electrical and Computer Engineering*, vol. 10, no. 2, pp. 30–34, 2010.
- [12] S. Agarwal and A. Goyal, "An automated approach for video surveillance using kalman filter," *International Journal on Recent and Innovation Trends in Computing and Communication ISSN*, vol. 4, no. 8, 2016.
- [13] J. Qranfal and C. L. Byrne, "Smart filter for dynamic SPECT image reconstruction," *International Journal of Pure and Applied Mathematics*, vol. 73, no. 4, pp. 69–78, 2011.
- [14] J. Qranfal and C. Byrne, "EM filter for time-varying SPECT reconstruction," *International Journal of Pure and Applied Mathematics*, vol. 73, no. 4, [2012 on issue title page], pp. 379–403, 2011.
- [15] S. Hu, S. Xu, D. Wang, and A. Zhang, "Optimization algorithm for kalman filter exploiting the numerical characteristics of SINS/GPS integrated navigation systems," *Sensors*, vol. 15, no. 11, pp. 28402–28420, 2015.
- [16] S. Pimentel and Y. Qranfal, "Data Assimilation with Filtering Methods that Use Kullback-Leibler Distance," 2016, <http://www.isda2016.net/assets/posters/PimentelDataassimilation.pdf>.
- [17] J. Humpherys, P. Redd, and J. West, "A fresh look at the Kalman filter," *SIAM Review*, vol. 54, no. 4, pp. 801–823, 2012.
- [18] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Printice-Hall, 1979.
- [19] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*, Wiley-Interscience, 2006.
- [20] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, Series D, Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [21] C. L. Byrne, "Iterative image reconstruction algorithms based on cross-entropy minimization," *IEEE Transactions on Image Processing*, vol. 2, no. 1, pp. 96–103, 1993.
- [22] C. L. Byrne, "Alternating Minimization and Alternating Projection Algorithms: A Tutorial," 2011, <http://faculty.uml.edu/cbyrne/CandT.pdf>.
- [23] I. Csiszar and G. Tusnady, "Information geometry and alternating minimization procedures," *Statistics & Decisions*, vol. 2, pp. 205–237, 1984.
- [24] Y. Vardi, L. A. Shepp, and L. Kaufman, "A statistical model for positron emission tomography," *Journal of the American Statistical Association*, vol. 80, no. 389, pp. 8–37, 1985.
- [25] L. Landweber, "An iteration formula for Fredholm integral equations of the first kind," *American Journal of Mathematics*, vol. 73, pp. 615–624, 1951.