

AN TIME-VARYING KALMAN FILTER APPLIED TO MOVING TARGET TRACKING

Nicolas Obolensky, Deniz Erdogmus, Jose C. Principe

*CNEL, Dept. of Electrical & Computer Engineering,
University of Florida, Gainesville, FL 32611*

Abstract: In this paper, we describe a time varying extension to the Kalman Filter for tracking moving objects in images. Classical methods for tracking assume either a constant model, in particular a constant acceleration, or a random acceleration. Those assumptions often turn out to be inaccurate. The proposed Kalman Filter adapts its model (i.e. the state transition matrix) at each step to better estimate the movement of the maneuvering target. The performance of this time-varying Kalman Filter is tested on an airborne surveillance video. *Copyright © Controlo 2002*

Keywords: Kalman Filter, adaptive, tracking.

1. INTRODUCTION

The Kalman filter has made a dramatic impact on linear estimation because of its modeling power and implementation simplicity for online estimation. Today, the Kalman filter is an established technique widely applied in the fields of navigation, guidance, aircraft and missile tracking, reentry of space vehicles, etc. For target tracking in video images, two main assumptions are made in the Kalman filter formulation: a constant state model with the acceleration as a state, often using position and rate measurements, as in [1,2]; or a constant state model with the acceleration considered as zero mean random input, as Friedland described in [3]. In the first model, the acceleration is considered to be constant. In that case, the vehicle velocity remains constant for a zero acceleration or grows towards infinity in the direction of the acceleration. The alternative method to this unrealistic model is a random acceleration. But the Kalman filter, when the acceleration is an input noise, will eventually cancel out a zero mean acceleration as an input to the state equations. In that method also, the vehicle velocity will tend to be constant. In any practical model to track moving objects in images, the velocity, as well as the acceleration, should be able to vary. Still, the acceleration variations for maneuvering targets may be considered smooth. It is therefore conceivable to

predict the current acceleration given the previous accelerations. In that case, the accelerations are no longer an input to the system, but become states as suggested in [4]. At the same time, however, it is possible to estimate the acceleration at each step given the three previous (noisy) position measurements. This acceleration estimate will allow us to adapt the state transition matrix during operation and therefore obtain a time varying model. Consequently, the Kalman filter model becomes time varying and will be adapted to the maneuvering target movement.

The organization of this paper is as follows. In Sec. 2, we present the structure of the adaptive Kalman filter. In Sec. 3, we study the performance of the proposed algorithm on a set of vehicle surveillance videos.

2. TIME-VARYING KALMAN FILTER

The maneuvering target movement can be described by the dynamic equations

$$\begin{aligned}x[k+1] &= A[k]x[k] + Bu[k] \\ y[k] &= Cx[k] + Du[k]\end{aligned}\quad (1)$$

where $x[k]$ is the state vector, $u[k]$ is the input vector and $y[k]$ is the output. $x[k]$ will be defined

as: $x[k] = \begin{bmatrix} p[k] \\ v[k] \\ a[k] \end{bmatrix}$, where $p[k]$ is the position of the

moving vehicle, $v[k]$ is its velocity and

$$\underline{a[k]} = \begin{bmatrix} a[k] \\ a[k-1] \\ \vdots \\ a[k-N] \end{bmatrix} \quad N \geq 2, \text{ with } a[i] \text{ being the}$$

acceleration at step i .

In our formulation there is no input to the system ($u[k]=0$) because the accelerations are considered as states.

2.1 The Adaptive Model

We have decided to work on the accelerations because they are the only possible innovation on the moving vehicle model.

We also assumed that the acceleration varies with time (sample time T), because, as we already said, a constant acceleration would make the velocity either constant or eventually infinite.

The second major assumption is that the current acceleration depends smoothly on the previous accelerations. If the number of previous accelerations is not too large (recent past), we furthermore assume that the acceleration can be modeled as a weighted sum of previous accelerations. At each step, we do a linear fitting of the acceleration data.

Since there is noise in our estimation of the acceleration, we consider that the acceleration vector is corrupted by additive Gaussian noise $n[k]$, with zero mean and covariance matrix $Q[k]$ to be estimated in the design.

$$\begin{cases} x[k+1] = x[k] + Tv[k] \\ v[k+1] = v[k] + Ta[k] \\ a[k+1] = w_{1,1}[k+1]a[k] + w_{1,2}[k+1]a[k-1] + \dots + w_{1,N}[k+1]a[k-N] + n_1[k] \\ a[k] = w_{2,1}[k+1]a[k-1] + w_{2,2}[k+1]a[k-2] + \dots + w_{2,N-1}[k+1]a[k-N] + n_2[k] \\ \vdots \\ a[k-N+1] = w_{N+1,1}[k+1]a[k-N] + n_{N+1}[k] \end{cases} \quad (2)$$

In a vector-matrix form, the dynamic equations (2) can be written as:

$$x[k+1] = A[k]x[k] + n[k] \quad (3)$$

$$\text{where } x[k] = \begin{bmatrix} p[k] \\ v[k] \\ a[k] \end{bmatrix},$$

$$A[k] = \begin{bmatrix} 1 & T & 0 & \dots & \dots & 0 \\ 0 & 1 & T & 0 & \dots & 0 \\ \vdots & 0 & w_{1,1}[k+1] & w_{1,2}[k+1] & \dots & w_{1,N}[k+1] \\ \vdots & \vdots & 0 & w_{2,1}[k+1] & W & w_{2,N}[k+1] \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & w_{N+1,1}[k+1] \end{bmatrix} \text{ and}$$

$$n[k] = \begin{bmatrix} 0 \\ 0 \\ n_1[k] \\ n_2[k] \\ \vdots \\ n_{N+1}[k] \end{bmatrix}.$$

The measurement equation can be written as:

$$y[k] = Cx[k] + v[k] \quad (4)$$

where $C = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & \dots & 1 \end{bmatrix}$ and $n[k]$ is a zero mean noise with a constant covariance matrix R .

The measurement are $p[k]$, the vehicle position at time step k and $a[k-2]$ the acceleration at time step k , given by $a[k-2] = \frac{2}{T^2}(p[k] - 2p[k-1] + p[k-2])$.

Notice that if there is a measurement error, the error on $a[i]$ is correlated to the error on $a[i-1]$, and therefore the noise is not exactly white. But this only changes the shape of the noise covariance matrices from diagonal matrices to symmetric Toeplitz matrices with all zero diagonals except the first principal one. Only the correlation between the $a[k-N]$ and $a[k-N-1]$ errors cannot be represented in the model, which should not alter the algorithm performance.

2.1 A Time-Varying Kalman Filter

Several time-varying Kalman filter models have been proposed, especially for visual robot tracking tasks [5]. Our new time-varying Kalman filter formulation is divided into three steps.

- *Model Update*

In practice the acceleration variations are smooth, so the assumed weighted linear dependence will not change drastically every step. Therefore, we use an on-line method to update the weights.

Each weight vector $w_1[k], w_2[k], \dots, w_{N+1}[k]$ is updated using the LMS algorithm.

$$w_i[k] = w_i[k-1] + \eta(\hat{a}[k-i+1] - w_i[k-1]a_{i,k})a_{i,k} \quad (5)$$

where $a_{i,k} = \begin{bmatrix} a[k-i] \\ \vdots \\ a[k-N+1] \end{bmatrix}$ is the input vector to the

Kalman filter at step $k-1$ and the desired response $\hat{a}[k-i+1]$ is the acceleration at the output of Kalman filter at the same step. So the error $e_i[k-1] = \hat{a}[k-i+1] - w_i[k-1]a_{i,k}$ is the difference between the optimal acceleration computed by the Kalman filter and the acceleration estimated by the LMS algorithm. We can also write the *Model Update* with the following formula:

$$A[k]=A[k-1]+\eta \begin{bmatrix} 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & e_1[k-1]\mathbf{a}_{1,k} & & \\ & & 0 & e_2[k-1]\mathbf{a}_{2,k} & \\ \vdots & \vdots & \vdots & 0 & \\ 0 & 0 & 0 & 0 & \ddots \end{bmatrix}$$

The step size η has to be chosen very carefully because it can make the whole algorithm fail. Too big a step size would make the model diverge, whereas too small a step size would prevent the model to adapt. Therefore, the approach for choosing η properly is to make it dependent on the system dynamics. The dynamics of the Kalman filter are represented by the matrix $(A[k] - K[k]C)$, where $K[k]$ is the Kalman gain at step k . The step size η will consequently be proportional to this matrix norm:

$$\eta[k] = \alpha \|A[k] - K[k]C\|$$

The proportionality coefficient α will be chosen according to the amplitude of the vehicle movement.

Then a classical Kalman Filter computes the output using the previously calculated model.

- *Time Update*

$$\begin{cases} \hat{x}[k] = A[k]\hat{x}[k-1] \\ P^-[k] = A[k]P[k-1]A^T[k] + Q[k] \end{cases} \quad (6)$$

- *Measurement Update*

$$\begin{cases} K[k] = P^-[k]C^T (CP^-[k]C^T + R)^{-1} \\ \hat{x}[k] = \hat{x}^-[k] + K[k](y[k] - C\hat{x}^-[k]) \\ P[k] = (I - K[k]C)P^-[k] \\ R[k] = CP[k]C^T \end{cases} \quad (7)$$

- *State Error Covariance Update*

The state error covariance update is useful to make sure that the *a priori* error covariance estimate in the Kalman filter the state noise coming from the LMS estimation.

$$Q[k] = \begin{bmatrix} 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ \vdots & \vdots & (e_1[k-1])^2 & 0 & \dots & 0 \\ \vdots & \vdots & 0 & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \dots & (e_{N+1}[k-1])^2 \end{bmatrix}$$

and finally the optimal estimates of the position $p[k]$ and the accelerations $a[k-1], a[k-2], \dots, a[k-N+1]$ are given by

$$y[k] = C\hat{x}[k] \quad (8)$$

3. RESULTS

The performance of this adaptive Kalman filter has been tested on a set of surveillance videos. The

detection part of the tracking method has been done manually. The output of this detection will be considered as the ground truth. The algorithm tracked only one vehicle on the ground, i.e. there is only two tracked coordinates x and y , the altitude not being considered.



Figure 1: test video example

To evaluate the performance of the adaptive Kalman filter, a noise with a constant variance has been added to the ground truth. The performance criteria are the errors on the position $p[k]$ and acceleration $a[k-2]$ measurements (we have considered $N=2$ in the $a[k]$ vector). They have been computed for different variances on the whole set of videos.

The next plot shows the results for a detection noise standard deviation of 5 pixels.

We represented separately the plot for x (on the left) and y (on the right). The first row shows the standard deviations of the position errors in pixels versus the time steps, the dashed line being the algorithm error standard deviation and the plain line being the measurement error standard deviation (that should be around 5, for this case). The second row represents the standard deviation of the pixel error between the real acceleration $a[k-2]$ and the acceleration computed by the proposed algorithm.

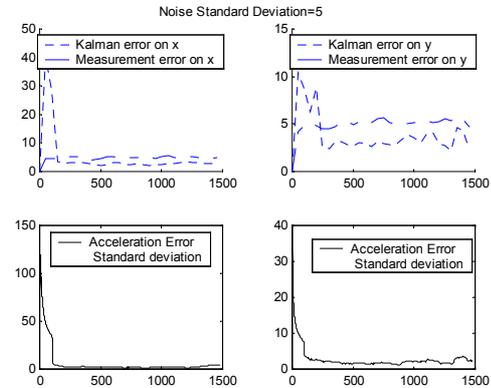


Figure 2: Acceleration and Position Error Standard deviations on x and y , with a detection noise standard deviation of 5, for the 1st video.

We first observe that the position errors go down for both x and y. After 250 steps, the standard deviation of the filter output error goes beneath that of the measured position error. As expected, the measured position error standard deviation is around 5 pixels, whereas after 1500 steps, the filter output error standard deviation goes below 3 pixels. At the same time, the acceleration error standard deviation for x and y decreases quickly to around 3 pixels per second², despite large errors at the beginning.

The next figure shows the results for another video with the same detection noise standard deviation of 5 pixels.

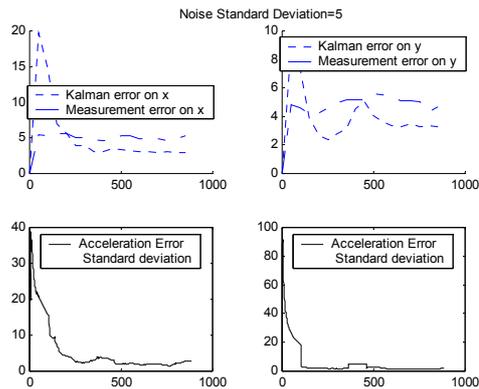


Figure 3: Acceleration and Position Error Standard deviations on x and y, with a detection noise standard deviation of 5, for the 2nd video.

For this other video, the filter output error also goes below the measurement error after 200 steps, when that acceleration error starts to be negligible.

The third plot shows the same parameters on the first video but for a detection noise standard deviation of 10 pixels.

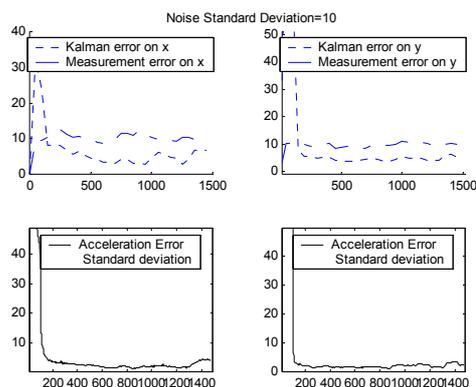


Figure 4: Acceleration and Position Error Standard deviations on x and y, with a detection noise standard deviation of 10, for the 1st video.

Even with a higher noise standard deviation, the observations remain the same. The filter output error, here around 5 pixels, is always below the measurement error, and the acceleration error is negligible.

In every experiment, the acceleration and the position errors are always inferior to the acceleration and position measurement. Besides, the algorithm becomes efficient after 200-250 steps, which corresponds for those MPEG videos to approximately 10 seconds of tracking.

Nevertheless, the convergence of this algorithm very much depends on the step size η . A too large step size will indeed make the accelerations grow exponentially and make the Kalman filter fail. Further work needs to be done to estimate the right step size for each problem.

Also, in all the experiments, the added detection noise is chosen to be an additive Gaussian noise, which is not necessarily the case. A classic detection algorithm needs to be applied to the data to make the performance evaluation of this time-varying Kalman filter more relevant.

6. CONCLUSIONS

In this paper, we have introduced a time-varying Kalman filter applied to moving vehicle tracking. The key idea is to consider the accelerations not as inputs to the system, but as states, in the model. Besides, in that model, the accelerations are modeled as a weighted sum of the previous accelerations. A linear fitting is recursively performed using an LMS algorithm. The computed model (state transition matrix, state error covariance matrix) is then used in a classic Kalman filter that has the vehicle position and several accelerations as its outputs. We have demonstrated the effectiveness of this filter on a set of airborne vehicle surveillance videos. After approximately 10 seconds of tracking, the algorithm becomes efficient, independently of the detection noise. However, further work needs to be done in order for the algorithm to be more robust, with a proper tuning of the step size, and for its performance to be independent of the detection noise characteristics.

This time-varying Kalman filter shows promising results for moving vehicles tracking.

Acknowledgments: This work was partially supported by a Honeywell Corporation grant.

REFERENCES

- [1] Ramachandra K.V., Mohan B.R., Geetha B.R. (1993). A Three-State Kalman Tracker Using Position and Rate Measurements. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no.1, pp. 215-222.
- [2] Ramachandra K.V. (1988). State estimation of manoeuvring targets from noisy radar measurements, *IEEE Proceedings*, vol.135, Pt.F,no.1 (Feb., 1988), pp. 82-84.
- [3] Friedland B. (1980). Steady state behaviour of Kalman filter with discrete- and continuous-time

observations, *IEEE Transactions on Automatic Control*, **AC-25**, (Oct., 1980), pp. 988-992.

- [4] Singer, R.A (1970). Estimating tracking filter performance for manned maneuvering targets, *IEEE Transactions on Aerospace and Electronic Systems*, **AES-6** (July, 1970), pp. 473-483.
- [5] Wira P., Urban J.P. (2000). A New Adaptive Kalman Filter Applied to Visual Servoing Tasks, *Proceedings of the International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, United Kingdom (Aug., 2000), pp. 267-270.