Object tracking with time-delayed, OOSM using Iterative AS-PDA algorithm

Javad Haddadnia¹, Hamidreza Rabiee², Omid Rahmani Seryasat³

¹Associate Professor, Electrical and Computer Engineering Department, Hakim Sabzevari University & Center for Research of Advanced Medical Technologies, Sabzvar University of Medical Sciences, Sabzevar, Iran
²Department of Electrical and Computer Engineering, Hakim Sabzevari University, Sabzevar, Iran
³Department of Electrical and Computer Engineering, Hakim Sabzevari University, Sabzevar, Iran

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

In recent times, there has been an explosion in the use of object tracking technology in non-military applications. Object tracking algorithms have become an essential part of our daily lives. For example, GPS-based navigation is a daily tool of humankind. In this application a group of artificial satellites in outer space continuously locate the vehicles people drive and the object tracking algorithms within the GPS perform self-localization and enable us to enjoy a number of location based services, such as finding places of interest and route planning. In this paper we present a novel method for object tracking with time delayed using iterative AS-PDA algorithm, which significantly improve the efficiency for tracking motion objects.

KEYWORDS: AS-PDA, OOSM, AS-KF

1- INTRODUCTION

Object tracking using delayed, out-of-sequence measurements is a problem of growing importance due to an increased reliance on networked sensors interconnected via complex communication network architectures. In such systems, it is often the case that measurements are received out-of-time-order at the processing computer.

This problem has appeared in the literature under various names such as the out-of-sequence measurements (OOSM) problem (Blackman and Popoli, 1999; Bar-Shalom, 2000), the problem of tracking with random sampling and delays (Marcus, 1979; Hilton et al., 1993; Thomopoulos and Zhang, 1994), and the problem of incorporating random time delayed measurements (Ravn et al., 1998).

In this paper, we present a Bayesian solution to this problem and provide approximate, implementable algorithms for both cluttered and non-cluttered scenarios involving single and multiple time-delayed measurements. Under linear Gaussian assumptions, the Bayesian solution reduces to an augmented state Kalman filter (AS-KF) for scenarios devoid of clutter and an augmented state probabilistic data association (AS-PDA) for scenarios involving clutter with modified measurement equations and likelihoods.

2- Optimal Bayesian solution to the OOSM problem

2-1- Object dynamics, sensor measurement equations

Usually the object dynamics are modeled by a discrete-time state equation,

\[ x_{k+1} = Fx_k + w_{k+1} \]  

And sensor measurements are modeled by,

\[ y_k = Hx_k + v_k \]

Where, \( H \) is the sensor measurement matrix and \( F \) is the object transition matrix.

Denoting a “standard” measurement sequence, \( y^s = \{ y_1, y_2, ..., y_k \} \), the object “standard” tracking problem reduces to the problem of computing the conditional mean estimate of the object state,

\[ \hat{x}_{kk} = E(x_k | y^s) \]

And its associated error covariance,

\[ P_{kk} = E[(x_k - \hat{x}_{kk})(x_k - \hat{x}_{kk})^T | y^s] \]

*Corresponding Author: Javad Haddadnia, Associate Professor, Electrical and Computer Engineering Department, Hakim Sabzevari University & Center for Research of Advanced Medical Technologies, Sabzvar University of Medical Sciences, Sabzevar, Iran. Email: Haddadnia@sttu.ac.ir*
A key problem arising when dealing with multiple interconnected sensors with communication links is the time delay between the sensor and tracking computer. This problem can be defined as follows: when a measurement corresponding to time \( \tau \), expressed as

\[
\tau_k = H \cdot x(\tau) + v(\tau), \tau \leq k
\]  

arrives at time \( t_k \) after (3) and (4) have been computed, one faces the problem of updating the state estimate and its covariance with the delayed measurement (5), i.e., to compute

\[
\hat{x}_{k|k} = E(x_k|y^k, y_{\tau})
\]  

(6)

And,

\[
P_{k|k} = E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})'|y^k, y(\tau)]
\]  

(7)

\[
... \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \qua
\[ \beta_i(k) = P(\theta_i(k) = P(\theta_{i,k}, \theta_{j,k-1}, ..., \theta_{j,k-l} | y^k) \]

\( i, j, l \in \{0, 1, 2, ...\} \)

5- Simulation results

In this part several applicable instances for presented algorithms are compared and the function of each is characterized using root mean square error over 600 Monte Carlo runs.

Only one lag is presumed in algorithms and the performance of them appraised. After that, we open the picture wider and allow several delays to occur between algorithms.

Then a chaos situation is regarded in last instance and the performance of algorithm (OOSM) is evaluated.

It is noteworthy that the output of state filter can be chosen from first or last component. The former algorithm includes “Filtering” and uses suffix “G”, however the latter includes more smooth filter and utilize suffix “F”.

The simulation results show that the presented algorithm have far better performance than earlier presented algorithms.

![Figure2. Augmented state PDA and its computation structure](image)

**Figure2.** Augmented state PDA and its computation structure

![Figure3. Logic flowchart for the single-cycle iterative AS-PDA.](image)

**Figure3.** Logic flowchart for the single-cycle iterative AS-PDA.
Figures 3 and 4 show the simulation results for the scenario, where the performance of the Y-algorithm, VDAS-KF and AS-KF2 are compared over 500 runs. A computational load comparison for these algorithms is listed in Table 1 in terms of the number of floating point operations normalized to that of a standard Kalman filter.

![Figure 3](image)

![Figure 4](image)

**Table 1 - Computational comparison**

<table>
<thead>
<tr>
<th>$P_e$</th>
<th>Y-algorithm</th>
<th>VDAS-KF</th>
<th>AS-KF2-F</th>
<th>AS-KF2-S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>5.57</td>
<td>5.57</td>
</tr>
<tr>
<td>0.25</td>
<td>2.26</td>
<td>2.60</td>
<td>5.57</td>
<td>5.57</td>
</tr>
<tr>
<td>0.5</td>
<td>2.30</td>
<td>2.68</td>
<td>5.57</td>
<td>5.57</td>
</tr>
<tr>
<td>0.75</td>
<td>4.41</td>
<td>5.47</td>
<td>5.57</td>
<td>5.57</td>
</tr>
</tbody>
</table>

**Conclusion**

This chapter discusses a solution to the very practical problem of real-time object tracking. In a scenario where a large number of sensors are connected through a network, measurements are received in a delayed time for a number of reasons. Therefore, assumptions of measurement availability at a particular time are not ensured and there is a need to incorporate that measurement when it is received at some later time. This problem is addressed in this chapter through the proposal of a Bayesian formulation of the solution. Several major algorithms – AS-KF, ASPDA, etc. are presented along with some illustrative results.

**REFERENCES**