

## Research Article

# Global Systems for Mobile Position Tracking Using Kalman and Lainiotis Filters

Nicholas Assimakis<sup>1</sup> and Maria Adam<sup>2</sup>

<sup>1</sup> Department of Electronic Engineering, Technological Educational Institute of Central Greece, 3rd km Old National Road Lamia-Athens, 35100 Lamia, Greece

<sup>2</sup> Department of Computer Science and Biomedical Informatics, University of Thessaly, 2-4 Papasiopoulou Street, 35100 Lamia, Greece

Correspondence should be addressed to Maria Adam; [madam@dib.uth.gr](mailto:madam@dib.uth.gr)

Received 3 August 2013; Accepted 2 January 2014; Published 23 February 2014

Academic Editors: S. Bourennane and J. Marot

Copyright © 2014 N. Assimakis and M. Adam. 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.

We present two time invariant models for Global Systems for Mobile (GSM) position tracking, which describe the movement in  $x$ -axis and  $y$ -axis simultaneously or separately. We present the time invariant filters as well as the steady state filters: the classical Kalman filter and Lainiotis Filter and the Join Kalman Lainiotis Filter, which consists of the parallel usage of the two classical filters. Various implementations are proposed and compared with respect to their behavior and to their computational burden: all time invariant and steady state filters have the same behavior using both proposed models but have different computational burden. Finally, we propose a Finite Impulse Response (FIR) implementation of the Steady State Kalman, and Lainiotis filters, which does not require previous estimations but requires a well-defined set of previous measurements.

## 1. Introduction

The Global Positioning System (GPS) is the most popular positioning technique in navigation providing reliable mobile location estimates in many applications [1–4]. Thus wireless location systems offering reliable mobile location estimates have been studied by researchers and engineers over the past few years. Various techniques require one base station or at least two base stations or more than three base stations in order to determine the location of the user. The accuracy of the positioning results is affected by many interference sources as the signals propagate in the atmosphere. So, techniques were developed using filters to estimate the location of the user through the location information exchanged between the handset and the base station. Kalman filter has been used in the localization process [4–6], due to the following advantages mentioned in [5]: (a) Kalman filter [7–9] processes noisy measurements and so it can smooth out the effects of noise in the estimated state variables by integrating more information from reliable data more than unreliable

data and (b) Kalman filter allows the combination of measurements from different sources (locomotion data) and different times. Kalman filter was implemented for Global Systems for Mobile (GSM) position tracking in [5]: Kalman filter was used for tracking in two dimensions and it was stated that Kalman filter is very powerful due to its reliable performance, because it yielded enhanced position tracking results.

In this paper we extend the ideas in [5] in two fields: (a) by using two models for GSM position tracking, which describe the movement in  $x$ -axis and  $y$ -axis simultaneously or separately and (b) by using the Kalman filter and the Lainiotis filter [8, 10]. The paper is organized as follows. In Section 2, we present two time invariant models for Global Systems for Mobile (GSM) position tracking, which describe the movement in  $x$ -axis and  $y$ -axis. In Section 3, we present the time invariant filters: Kalman filter, Lainiotis Filter and Join Kalman Lainiotis Filter. In Section 4, we present the corresponding steady state filters. In Section 5, various implementations are proposed. In Section 6, we compare the filters with respect to their behavior and to their computational

burden. In Section 7, we propose a Finite Impulse Response (FIR) implementation of the Steady State Kalman and Lainiotis Filters. Finally, Section 8 summarizes the conclusions.

### 2. Time Invariant Models

Linear estimation is associated with time invariant systems described by the following state space equations:

$$\begin{aligned} x(k+1) &= Fx(k) + Gw(k), \\ z(k) &= Hx(k) + v(k) \end{aligned} \tag{1}$$

for  $k \geq 0$ , where  $x(k)$  is the  $n$ -dimensional state vector at time  $k$ ,  $z(k)$  is the  $m$ -dimensional measurement vector at time  $k$ ,  $F$  is the  $n \times n$  system transition matrix,  $H$  is the  $m \times n$  output matrix,  $w(k)$  is the plant noise at time  $k$ ,  $v(k)$  is the measurement noise at time  $k$ . Also,  $\{w(k)\}$  and  $\{v(k)\}$  are Gaussian zero-mean white random processes with covariance matrices  $Q$  and  $R$ , respectively. The initial state  $x(0)$  is a Gaussian random variable with mean  $x_0$  and covariance  $P_0$  and is assumed to be independent of  $w(k)$  and  $v(k)$ .

In this paper we consider two models.

*Model A.* The first model (model A) describes the movement in  $x$ -axis and  $y$ -axis simultaneously and follows the ideas in [5].

The state vector is of dimension  $n = 4$  and contains the position and the velocity in  $x$ -axis and  $y$ -axis:  $x(k) = [s_x(k) \ v_x(k) \ s_y(k) \ v_y(k)]^T$ . The measurement vector is of dimension  $m = 2$  and contains the measured position  $x$ -axis and  $y$ -axis:  $z(k) = [z_x(k) \ z_y(k)]^T$ .

Then we take:

$$F = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{2}\Delta t & 0 \\ 1 & 0 \\ 0 & \frac{1}{2}\Delta t \\ 0 & 1 \end{bmatrix}, \tag{2}$$

$$H = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & \Delta t \end{bmatrix}.$$

The plant noise  $w(k) = [w_x(k) \ w_y(k)]^T$  is Gaussian zero-mean with covariance matrix  $Q = \begin{bmatrix} \sigma_{xq}^2 & 0 \\ 0 & \sigma_{yq}^2 \end{bmatrix}$ .

The measurement noise  $v(k) = [v_x(k) \ v_y(k)]^T$  is Gaussian zero-mean with covariance matrix  $R = \begin{bmatrix} \sigma_{xr}^2 & 0 \\ 0 & \sigma_{yr}^2 \end{bmatrix}$ .

*Model B.* The second model (model B) describes the movement in  $x$ -axis and  $y$ -axis separately. In each axis, the state vector is of dimension  $n = 2$  and contains the position and the velocity:  $x(k) = [s(k) \ v(k)]^T$ . The measurement vector is of dimension  $m = 1$  and contains the measured position  $z(k)$ .

Then we take:

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} \frac{1}{2}\Delta t \\ 1 \end{bmatrix}, \quad H = [1 \ \Delta t]. \tag{3}$$

The plant noise  $w(k)$  is Gaussian zero-mean with covariance matrix  $Q = \sigma_q^2$ .

The measurement noise  $v(k)$  is Gaussian zero-mean with covariance matrix  $R = \sigma_r^2$ .

It is obvious that we are able to describe the movement in both axes using two separate state vectors:  $x_x(k) = [s_x(k) \ v_x(k)]^T$  for the  $x$ -axis and  $x_y(k) = [s_y(k) \ v_y(k)]^T$  for the  $y$ -axis. If we merge these two state vectors, we take the state vector  $x(k) = [s_x(k) \ v_x(k) \ s_y(k) \ v_y(k)]^T$  of model A.

### 3. Time Invariant Kalman and Lainiotis Filters

In this section, we present the classical time invariant Kalman filter [7–9] and Lainiotis Filter [8, 10], which are the most well-known algorithms that solve the filtering problem. Both algorithms compute the estimation  $x(k/k)$  and the corresponding estimation error covariance  $P(k/k)$ . We also propose the Join Kalman-Lainiotis Filter which consists of the parallel (with the same measurements) usage of *two* filters (one Kalman filter and one Lainiotis Filter) and combination of the results (weight 50% for each filter).

*Kalman Filter (KF).* The following equations constitute the KF:

$$\begin{aligned} x(k+1/k) &= Fx(k/k), \\ P(k+1/k) &= (GQG^T) + FP(k/k)F^T, \\ K(k+1) &= P(k+1/k)H^T[HP(k+1/k)H^T + R]^{-1}, \\ x(k+1/k+1) &= [I - K(k+1)H]x(k+1/k) \\ &\quad + K(k+1)z(k+1), \end{aligned} \tag{4}$$

$$P(k+1/k+1) = [I - K(k+1)H]P(k+1/k),$$

for  $k \geq 0$ , with initial conditions  $x(0/0) = x_0$  and  $P(0/0) = P_0$ .

The Kalman filter computes the estimation  $x(k/k)$  and the estimation error covariance  $P(k/k)$  through the prediction  $x(k+1/k)$  and the corresponding prediction error covariance  $P(k+1/k)$  using the Kalman filter gain  $K(k)$ .

*Lainiotis Filter (LF).* The following equations constitute the LF:

$$\begin{aligned} x(k+1/k+1) &= K_n z(k+1) + F_n [I + P(k/k)O_n]^{-1} \\ &\quad \times [P(k/k)K_m z(k+1) + x(k/k)], \end{aligned} \tag{5}$$

$$P(k+1/k+1) = P_n + F_n [I + P(k/k)O_n]^{-1} P(k/k) F_n^T \tag{6}$$

for  $k \geq 0$ , with initial conditions  $x(0/0) = x_0$  and  $P(0/0) = P_0$ , where

$$\begin{aligned} A &= [H(GQG^T)H^T + R]^{-1}, \\ K_n &= (GQG^T)H^T A, \\ K_m &= F^T H^T A, \end{aligned}$$

$$\begin{aligned}
 P_n &= (GQG^T) - K_nH(GQG^T), \\
 F_n &= F - K_nHF, \\
 O_n &= F^T H^T AHF.
 \end{aligned}
 \tag{7}$$

*Join Kalman-Lainiotis Filter (JKLF).* The filter consists of the parallel usage of *two* filters (one Kalman filter and one Lainiotis Filter) with the same measurements and combination of the results (weight 50% for each filter):

$$\begin{aligned}
 x(k/k) &= \frac{1}{2}x_{KF}(k/k) + \frac{1}{2}x_{LF}(k/k), \\
 P(k/k) &= \frac{1}{2}P_{KF}(k/k) + \frac{1}{2}P_{LF}(k/k).
 \end{aligned}
 \tag{8}$$

#### 4. Steady State Kalman and Lainiotis Filters

For time invariant systems, it is well known [7] that there exists a steady state value  $P_p$  of the prediction error covariance matrix, if the signal process model is asymptotically stable, or if the signal process model is not necessarily asymptotically stable, but the pair  $[F, H]$  is completely detectable and the pair  $[F, GG_1]$  is completely stabilizable for any  $G_1$  with  $G_1G_1^T = Q$ . Then there also exist a steady state value  $P_e$  of the estimation error covariance matrix and a steady state value  $K$  of the Kalman filter gain.

In this section we present the Steady State Kalman filter and Lainiotis Filter. Both algorithms compute the estimation  $x(k/k)$  using the previous estimation and the current measurement. We also propose the Join Steady State Kalman-Lainiotis Filter, which consists of the parallel usage of *two* filters (one Steady State Kalman filter and one Steady State Lainiotis Filter) with the same measurements and combination of the results (weight 50% for each filter).

*Steady State Kalman Filter (SSKF).* The following equation constitutes the SSKF:

$$x(k+1/k+1) = A_{KF}x(k/k) + B_{KF}z(k+1) \tag{9}$$

for  $k \geq 0$ , with initial condition  $x(0/0) = x_0$ , where

$$\begin{aligned}
 A_{KF} &= [I - KH]F, \\
 B_{KF} &= K.
 \end{aligned}
 \tag{10}$$

The steady state Kalman filter gain  $K$  is computed by  $K = P_p H^T [HP_p H^T + R]^{-1}$ , where  $P_p$  is the steady state prediction error covariance computed by solving the *Riccati equation emanating from Kalman filter (REKF)*:

$$P_p = (GQG^T) + FP_p F^T - FP_p H^T [HP_p H^T + R]^{-1} HP_p F^T. \tag{11}$$

In view of the importance of the Riccati equation emanating from Kalman filter, there exists considerable literature on its algebraic solutions [7, 11] or iterative solutions [7, 12–15] concerning per step or doubling algorithms.

*Steady State Lainiotis Filter (SSLF).* The following equation constitutes the SSLF:

$$x(k+1/k+1) = A_{LF}x(k/k) + B_{LF}z(k+1), \tag{12}$$

for  $k \geq 0$ , with initial condition  $x(0/0) = x_0$ , where

$$\begin{aligned}
 A_{LF} &= F_n [I + P_e O_n]^{-1}, \\
 B_{LF} &= K_n + F_n [I + P_e O_n]^{-1} P_e K_m,
 \end{aligned}
 \tag{13}$$

and  $P_e$  is the steady state estimation error covariance computed by solving the *Riccati equation emanating from Lainiotis filter (RELf)*:

$$P_e = P_n + F_n [I + P_e O_n]^{-1} P_e F_n^T. \tag{14}$$

In view of the importance of the Riccati equation emanating from Lainiotis Filter, there exists considerable literature on its algebraic or iterative solutions [12, 14–16] concerning per step or doubling algorithms.

Note that in [8] it is shown that SSKF is equivalent to SSLF, since

$$\begin{aligned}
 A_{KF} &= A_{LF}, \\
 B_{KF} &= B_{LF}.
 \end{aligned}
 \tag{15}$$

*Join Steady State Kalman-Lainiotis Filter (JSSKLF).* The filter consists of the parallel usage of *two* steady state filters (one Steady State Kalman filter and one Steady State Lainiotis Filter) with the same measurements and combination of the results (weight 50% for each filter):

$$x(k/k) = \frac{1}{2}x_{KF}(k/k) + \frac{1}{2}x_{LF}(k/k), \tag{16}$$

for  $k \geq 0$ .

#### 5. Implementations

In this section, we propose various implementations.

The use of model A which describes the movement in  $x$ -axis and  $y$ -axis simultaneously requires the use one filter; we are able to use KF/LF/SSKF/SSLF/JKLF in order to compute the estimation and the corresponding estimation error covariance.

The use of model B, which describes the movement in  $x$ -axis and  $y$ -axis separately, requires the use of two filters KF/LF/SSKF/SSLF/JSSKLF in order to compute the estimation and the corresponding estimation error covariance for each movement. It is obvious that, if we merge the estimation  $x_x(k/k) = [s_x(k/k) \ v_x(k/k)]^T$  for the movement in  $x$ -axis and the estimation  $x_y(k/k) = [s_y(k/k) \ v_y(k/k)]^T$  for the movement in  $y$ -axis, we take the state vector of model A:

$$\begin{aligned}
 x(k/k) &= [s_x(k/k) \ v_x(k/k) \ s_y(k/k) \ v_y(k/k)]^T \\
 &= [x_x(k/k) \ x_y(k/k)]^T.
 \end{aligned}
 \tag{17}$$

TABLE 1: GSM position tracking implementations.

Implementation	Model	System	Filter
1			KF
2		Time invariant	LF
3	Model A		JKLF
4			SSKF
5		Steady state	SSLF
6			JSSKLF
7			KF
8		Time invariant	LF
9	Model B		JKLF
10			SSKF
11		Steady state	SSLF
12			JSSKLF

Also, the estimation error covariances  $P_x(k/k)$  and  $P_y(k/k)$  for each movement can be merged to the estimation error covariance of model A:

$$P(k/k) = \begin{bmatrix} P_x(k/k) & 0 \\ 0 & P_y(k/k) \end{bmatrix}. \quad (18)$$

Thus, we propose various implementations for Global Systems for Mobile (GSM) position tracking, as it is shown in Table 1.

## 6. Comparison of the Filters

In this section we compare the filters with respect to their behavior and to their computational burden.

*Example 1.* We implemented the filters with the following parameters:

- (i) discretization factor:  $\Delta t = 1$ ,
- (ii) movement reliability:  $\sigma_{xq}^2 = \sigma_{yq}^2 = 0.01$ ,
- (iii) measurements reliability:  $\sigma_{xr}^2 = \sigma_{yr}^2 = 0.1$ ,
- (iv) initial conditions:  $x_0 = 0$  and  $P_0 = 0$ .

Concerning the behavior of the filters, we found that

- (i) the time invariant filters KF, LF and JKLF are equivalent, since they compute the same outputs (estimation and estimation error covariance), using model A or model B,
- (ii) the steady state filters SSKF, SSLF and JSSKLF are equivalent, since they compute the same outputs (estimation and estimation error covariance), using model A or model B,
- (iii) the steady state filters and the time invariant filters compute outputs very close to each other,
- (iv) model A is equivalent to model B, since they produce the same outputs.

These results are depicted in Figure 1.

Concerning the computational burden of the filters, we compared the filters with respect to their per-iteration calculation burdens, computed using the ideas in [8], as shown in Table 2.

Table 3 summarizes the per-iteration calculation burden of all implementations, using model A and model B.

We observe that

- (i) KF is faster than LF,
- (ii) JKLF is slower than KF and LF (since the join filter requires the implementation of both the Kalman and Lainiotis filters),
- (iii) SSKF is as fast as SSLF,
- (iv) SSKF and SSLF are faster than KF and LF,
- (v) JSSKLF is slower than SSKF and SSLF,
- (vi) the filters using model B are faster than the same filters using model A.

Table 4 summarizes speedup between the various implementations.

We observe that

- (i) KF is faster than LF,

$$\begin{aligned} \text{speedup (LF model A to KF model A)} &= 1.316, \\ \text{speedup (LF model B to KF model B)} &= 1.290. \end{aligned}$$

- (ii) Model B is faster than model A,

$$\begin{aligned} \text{speedup (KF model A to SSKF model B)} &= 25.450, \\ \text{speedup (LF model A to SSLF model B)} &= 33.500. \end{aligned}$$

## 7. FIR Steady State Kalman and Lainiotis Filters

In this section we propose an FIR implementation of the Steady State Kalman filter and the Steady State Lainiotis Filter.

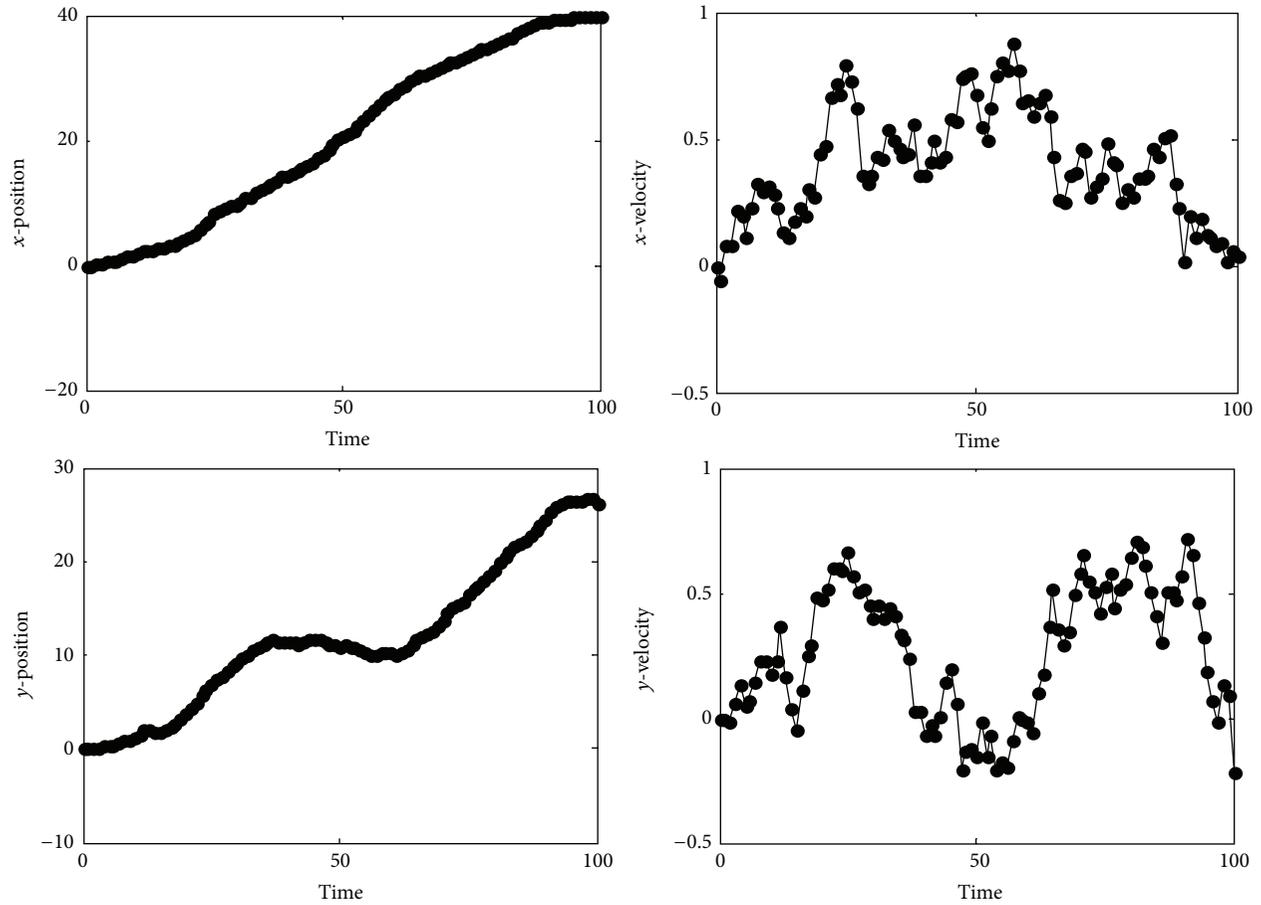


FIGURE 1: Position and velocity estimation solid line: KF/LF/JKLF, dashed line: SSKF/SSLF/JSSKLF.

TABLE 2: Per-iteration calculation burden of filters.

KF	$4n^3 + 3.5n^2 - 1.5n + 4n^2m + nm + 3nm^2 + (16m^3 - 3m^2 - m)/6$
LF	$4nm + (58n^3 + 9n^2 - 7n)/6$
JKLF	$n^2 + 3n$ (join procedure)
SSKF	$2n^2 + 2nm - n$
SSLF	$2n^2 + 2nm - n$
JSSKLF	$2n$ (join procedure)

TABLE 3: Per-iteration calculation burden of implementations.

Implementation	Model	System	Filter	Calculation burden
1	Model A	Time invariant	KF	509
2			LF	670
3			JKLF	1207
4		Steady state	SSKF	44
5			SSLF	44
6			JSSKLF	96
7	Model B	Time invariant	KF	138
8			LF	178
9			JKLF	326
10		Steady state	SSKF	20
11			SSLF	20
12			JSSKLF	44

TABLE 4: Speedup.

	1	2	3	Model A		5	6	7	8	9	Model B		11	12
	KF	LF	JKLF	SSKF	SSLF	JSSKLF	KF	LF	SSKF	SSLF	JSSKLF	SSKF	SSLF	JSSKLF
1	KF	0,760	0,427	11,568	11,568	4,314	3,688	2,860	25,450	0,795	25,450	25,450	25,450	5,194
2	LF	1,316	0,563	15,227	15,227	5,678	4,855	3,764	33,500	1,047	33,500	33,500	33,500	6,837
3	JKLF	2,340	1,778	27,068	27,068	10,093	8,630	6,691	59,550	1,861	59,550	59,550	59,550	12,153
4	SSKF	0,086	0,066	0,037	1,000	0,373	0,319	0,247	2,200	0,069	2,200	2,200	2,200	0,449
5	SSLF	0,086	0,066	0,037	1,000	0,373	0,319	0,247	2,200	0,069	2,200	2,200	2,200	0,449
6	JSSKLF	0,232	0,176	0,099	2,682	0,831	0,855	0,663	5,900	0,184	5,900	5,900	5,900	1,204
7	KF	0,271	0,116	3,136	3,136	1,169	1,290	0,775	6,900	0,216	6,900	6,900	6,900	1,408
8	LF	0,350	0,149	4,045	4,045	1,508	1,290	0,775	8,900	0,278	8,900	8,900	8,900	1,816
9	JKLF	1,257	0,955	0,537	14,545	5,424	4,638	3,596	32,000	0,031	32,000	32,000	32,000	6,531
10	SSKF	0,039	0,030	0,017	0,455	0,169	0,145	0,112	1,000	0,031	1,000	1,000	1,000	0,204
11	SSLF	0,039	0,030	0,017	0,455	0,169	0,145	0,112	1,000	0,031	1,000	1,000	1,000	0,204
12	JSSKLF	0,193	0,146	0,082	2,227	0,831	0,710	0,551	4,900	0,153	4,900	4,900	4,900	0,204

TABLE 5: Calculation burden of classical and FIR Steady State filters.

SSKF/SSLF	$2n^2 + 2nm - n$	Per-iteration
FIR SSKF/SSLF	$2n^2 + 2nm - n + (M - 1)n$	When required

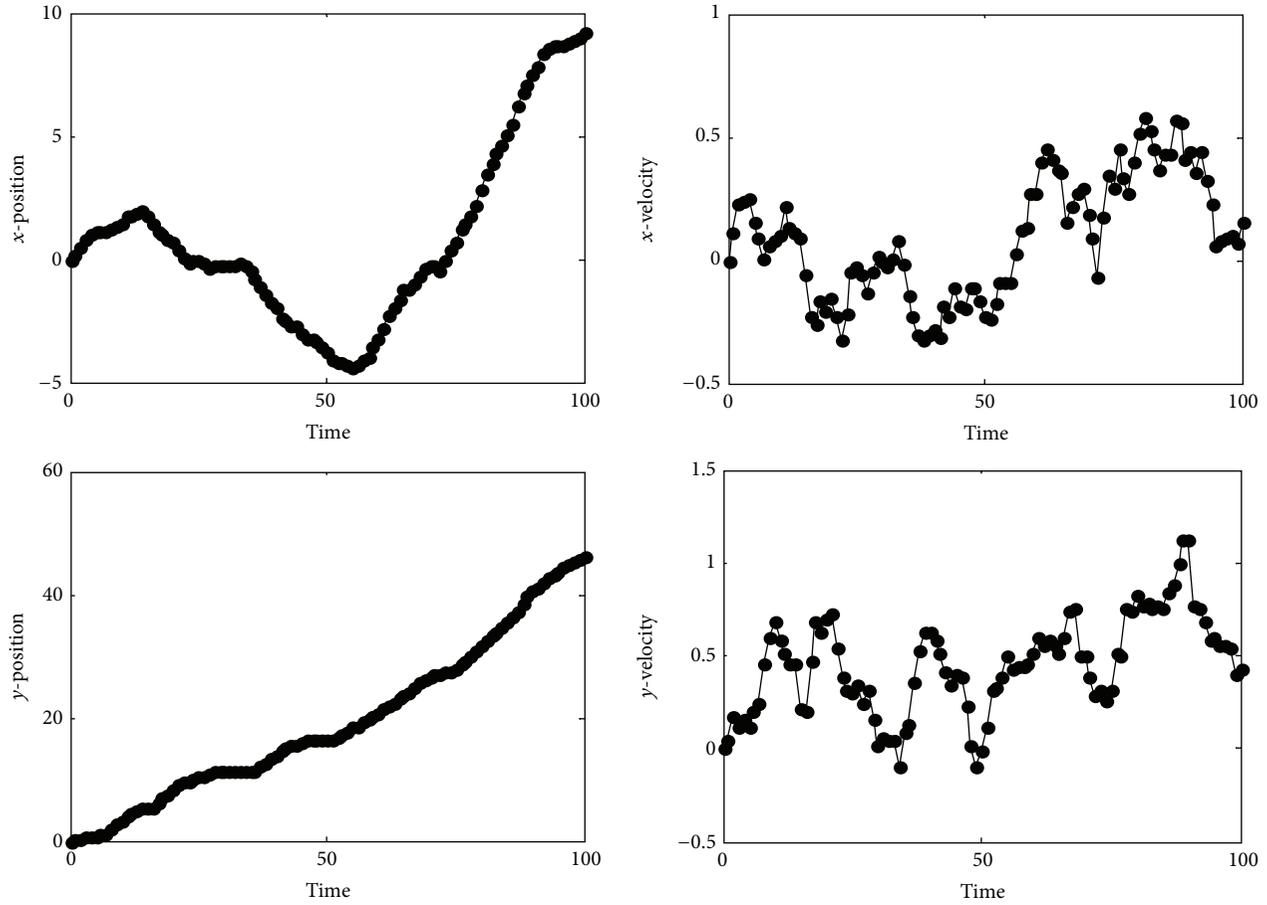


FIGURE 2: Position and velocity estimation solid line: SSKF/SSLF, dashed line: FIR SSKF/SSLF.

Recall that SSKF and SSLF have equal parameters:

$$A = A_{KF} = A_{LF}, \quad B = B_{KF} = B_{LF}. \quad (19)$$

Then we are able to write

$$x(k+1/k+1) = Ax(k/k) + Bz(k+1), \quad (20)$$

for  $k \geq 0$  with initial condition  $x(0/0) = x_0$ .

Then we take:

$$x(1/1) = Ax_0 + Bz(1)$$

$$x(2/2) = A^2x_0 + ABz(1) + Bz(2)$$

⋮

$$x(\nu/\nu) = A^\nu x_0 + A^{\nu-1}Bz(1) + \dots + ABz(\nu-1) + Bz(\nu). \quad (21)$$

Using the ideas in [17], the resulting FIR SSKF/SSLF is formulated as

$$x(k/k) = \sum_{j=0}^{M-1} A^j Bz(k-j), \quad \text{for } k \geq 1, \quad (22)$$

where  $A = A_{KF} = A_{LF}$  and  $B = B_{KF} = B_{LF}$  and  $M$  is the FIR filter order defined by  $\|A^M\| < \varepsilon$  and  $\|A^{M-1}\| \geq \varepsilon$ , with  $\varepsilon$  a small real value.

*Remarks.* (1) The FIR steady state filter coefficients can be calculated off-line by solving the corresponding Riccati equation.

(2) The FIR steady state filter does not require previous estimations but it requires a well-defined set of  $M$  previous measurements. This means that we have to wait for  $M$  time moments in order to produce the results. Alternatively, we are able to use only the available measurements until time  $M$  is reached or to use SSKF until time  $M$ .

TABLE 6: Calculation burden of classical and FIR steady state filters implementations.

	Model A	Model B
SSKF/SSLF	44	20
FIR SSKF/SSLF	$40 + 4M$	$16 + 4M$

TABLE 7: FIR results for Example 1.

$M = 17$	Model A	Model B
SSKF/SSLF	44	20
FIR SSKF/SSLF	108	84
$\mu$	3	5

We implemented the FIR filter for the parameters of the example in Section 6. We used  $\varepsilon = 10^{-3}$  and we found  $M = 17$ .

The steady state filters and FIR steady state filters compute outputs very close to each other, as depicted in Figure 2.

Concerning the computational burden, the FIR steady state filter possesses a constant burden while the classical steady state filter (SSKF/SSLF) possesses a constant per-iteration computational burden, as it is shown in Table 5.

Table 6 summarizes the calculation burden of the classical and FIR steady state filters implementations, using model A and model B.

Thus, FIR SSKF/SSLF can be faster than SSKF/SSLF, if we take results using FIR SSKF/SSLF every (or more than)  $\mu$  time lags, where  $\mu$  is the nearest integer greater than or equal to the ratio  $(40 + 4M)/44 = (10 + M)/11$  for model A and  $(16 + 4M)/20 = (4 + M)/5$  for model B.

For our example, we take the results, which are appeared in Table 7.

## 8. Conclusions

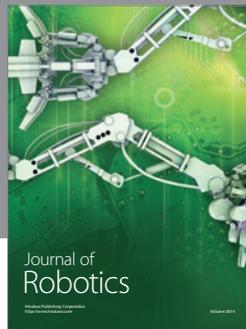
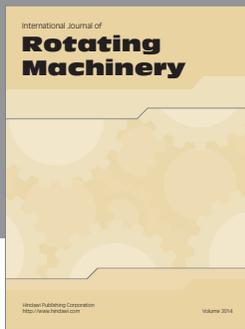
In this paper we presented two time invariant models for Global Systems for Mobile (GSM) position tracking, which describe the movement in  $x$ -axis and  $y$ -axis simultaneously or separately. We presented the time invariant filters as well as the steady state filters: the classical Kalman filter and Lainiotis Filter and the Join Kalman Lainiotis Filter, which consists of the parallel usage of the two classical filters. Various implementations are proposed and compared with respect to their behavior and to their computational burden. We found that all time invariant and steady state filters have the same behavior using both of the proposed models. We found that (a) Kalman filter is faster than Lainiotis Filter, (b) Join Kalman Lainiotis Filter is slower than both Kalman filter and Lainiotis Filter, (c) steady state filters are faster than time invariant filters and (d) the filters using the model, which handles the movement in  $x$ -axis and  $y$ -axis separately, are faster than the same filters using the model, which handles the movement in  $x$ -axis and  $y$ -axis simultaneously. Finally, we proposed an FIR implementation of the Steady State Kalman and Lainiotis Filters, which does not require previous estimations but it requires a well-defined set of previous measurements.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] C. Drane, M. Macnaughtan, and C. Scott, "Positioning GSM telephones," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 46–59, 1998.
- [2] A. Küpper, *Location-Based Services*, John Wiley and Sons, 1st edition, 2005.
- [3] M. Mouly and M. Pautet, *The GSM System for Mobile Communications*, Telecom Publishing, Washington, DC, USA, 1992.
- [4] S. Rooney, P. Chippendale, R. Choony, C. Le Roux, and B. Honary, "Accurate vehicular positioning using a DAB-GSM hybrid system," in *Proceedings of the 51st Vehicular Technology Conference*, pp. 97–101, May 2000.
- [5] J. P. Dubois, J. S. Daba, M. Nader, and C. El Ferkh, "GSM position tracking using a Kalman filter," *World Academy of Science, Engineering and Technology*, vol. 68, pp. 1610–1619, 2012.
- [6] T. Kos, M. Grgic, and G. Sisul, "Mobile user positioning in GSM/UMTS cellular networks," in *Proceedings of the 48th International Symposium ELMAR-2006 on Multimedia Signal Processing and Communications*, pp. 185–188, June 2006.
- [7] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Dover Publications, New York, NY, USA, 2005.
- [8] N. Assimakis and M. Adam, "Discrete time Kalman and Lainiotis filters comparison," *International Journal of Mathematical Analysis*, vol. 1, no. 13, pp. 635–659, 2007.
- [9] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering D*, vol. 82, no. 1, pp. 34–45, 1960.
- [10] D. G. Lainiotis, "Partitioned linear estimation algorithms: discrete case," *IEEE Transactions on Automatic Control*, vol. 20, no. 2, pp. 255–257, 1975.
- [11] D. R. Vaughan, "Nonrecursive algebraic solution for the discrete Riccati equation," *IEEE Transactions on Automatic Control*, vol. 15, no. 5, pp. 597–599, 1970.
- [12] N. D. Assimakis, D. G. Lainiotis, S. K. Katsikas, and F. L. Sanida, "A survey of recursive algorithms for the solution of the discrete time riccati equation," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 30, no. 4, pp. 2409–2420, 1997.
- [13] N. Assimakis, S. Roulis, and D. Lainiotis, "Recursive solutions of the discrete time Riccati equation," *Neural, Parallel and Scientific Computations*, vol. 11, pp. 343–350, 2003.
- [14] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, "A new computationally effective algorithm for solving the discrete Riccati equation," *Journal of Mathematical Analysis and Applications*, vol. 186, no. 3, pp. 868–895, 1994.
- [15] D. G. Lainiotis, N. D. Assimakis, and S. K. Katsikas, "Fast and numerically robust recursive algorithms for solving the discrete time Riccati equation: the case of nonsingular plant noise covariance matrix," *Neural, Parallel and Scientific Computations*, vol. 3, no. 4, pp. 565–584, 1995.
- [16] D. G. Lainiotis, "Discrete Riccati equation solutions: partitioned algorithms," *IEEE Transactions on Automatic Control*, vol. 20, no. 4, pp. 555–556, 1975.
- [17] N. Assimakis and M. Adam, "FIR implementation of the steady state Kalman filter," *International Journal of Signal and Imaging Systems Engineering*, vol. 1, no. 3-4, pp. 279–286, 2008.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

