ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 4, Issue 3, pp: (261-272), Month: July - September 2016, Available at: www.researchpublish.com

MOBILE POSITION ESTIMATION AND PREDICTION USING STEADY STATE KALMAN FILTER

N. Assimakis[1], G. Tziallas[2], M. Adam[3], A. Polyzos[4], C. Papanastasiou[5]

[1,2]Department of Electronic Engineering, Technological Educational Institute of Central Greece, 3rd km Old National Road Lamia-Athens, 35131 Lamia, Greece
[3]Department of Computer and Biomedical Informatics, University of Thessaly, 2-4 Papasiopoulou Street, 35131 Lamia, Greece
[5]Cross Software Solutions IKE, Leoforos Afentouli 2, Piraeus, 18536, Greece
[5]Department of Electrical Engineering, Technological Educational Institute of Central Greece, 3rd km Old National Road Lamia-Athens, 35131 Lamia, Greece

Abstract: **Time invariant models describing constant velocity and constant acceleration movements in two dimensions are presented. Mobile position tracking is realized for position estimation or prediction using steady state Kalman filter, concluding that the obtained estimates are very close to the real position.**

Keywords: **Kalman filter, steady state, position estimation, prediction.**

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 over the past few years. The location of the used is determined using one or more base stations. The position accuracy is affected by interference sources, leading to the need to develop techniques to estimate the location of the user. Kalman filter has been used for Global Systems for Mobile (GSM) position tracking in two dimensions [5]. Kalman and Lainiotis filters have been used in [6] where the GSM position tracking was derived using models that describe the movement in x-axis and y-axis simultaneously or separately. Also, Mobile Position Tracking in three dimensions using Kalman and Lainiotis filters is presented in [7].

In this paper estimation as well prediction algorithms are presented using the steady state Kalman filter. The novelty of the paper is the derivation of steady state estimation and prediction algorithms. It is shown that the position estimation and prediction are reliable. It is also shown that estimation calculation burden equals the prediction calculation burden; this steady state estimation/prediction calculation is much less than the calculation burden of the Kalman filter.

The paper is organized as follows: In Section 2, we present two models for Mobile Position Tracking (MPT), which describe constant velocity and constant acceleration movements in two dimensions. In Section 3, we present the steady state Kalman filter. In Section 4, we present steady state prediction algorithms. In Section 5, simulation results are presented. Finally, Section 6 summarizes the conclusions.

1.1 time invariant models:

In this paper we consider two models for Mobile Position Tracking (MPT), which describe constant velocity and constant acceleration movements in two dimensions. We assume continuous as well discrete process models to describe the constant velocity and the constant acceleration movements, where the primary element of the state is the position and the measurement is a noisy position.

1.2 general continuous models:

The General Continuous Model consists of the dynamic and the statistical model.

The dynamic model expresses the relationship between state and the measurement and is described by the following state space equations:

$$\dot{x}(t) = \Phi x(t) + Gw(t) \quad (1)$$

$$z(t) = \mathrm{H}x(t) + v(t) \quad (2)$$

where x(t) is the nx1 state vector, z(t) is the mx1 measurement vector, Φ is the nxn transition matrix, H is the mxn output matrix, w(t) is the state noise and v(t) is the measurement noise at time t.

The statistical model expresses the nature of the state and the measurements. Basic assumption is that the state and the measurement noises are Gaussian zero-mean white random processes with known covariance matrices Q and R, respectively. The initial state x(0) is a Gaussian random variable with mean x_0 and covariance P_0. Also, the state and the measurement noises and the initial state are independent.

2. GENERAL DISCRETE MODEL

The General Discrete Model consists of the dynamic and the statistical model.

The dynamic model expresses the relationship between state and the measurement and is described by the following state space equations:

$$x(k+1) = Fx(k) + w(k) \quad (3)$$

$$z(k) = Hx(k) + v(k) \quad (4)$$

where x(k) is the nx1 state vector, z(k) is the mx1 measurement vector, F is the nxn transition matrix, H is the mxn output matrix, w(k) is the state noise and v(k) is the measurement noise at time k.

The statistical model expresses the nature of the state and the measurements. Basic assumption is that the state and the measurement noises are Gaussian zero-mean white random processes with known covariance matrices Q and R, respectively. The initial state x(0) is a Gaussian random variable with mean x_0 and covariance P_0. Also, the state and the measurement noises and the initial state are independent.

2.1. Constant velocity model:

The constant velocity model describes the constant velocity movement in two dimensions separately. We start with the constant velocity movement in one dimension.

Continuous Model:

The velocity is constant. The state vector has two elements, the position and the velocity and the measurement vector has one element, the noisy position.

$$\dot{x}(t) = \Phi x(t) + Gw(t) \Rightarrow \begin{bmatrix} \dot{s}(t) \\ \dot{\upsilon}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ \upsilon(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t)$$

$$z(t) = \mathrm{H}x(t) + v(t) \Rightarrow z(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} s(t) \\ \upsilon(t) \end{bmatrix} + v(t)$$

Discrete Model:

The velocity is constant. The state vector has two elements, the position and the velocity and the measurement vector has one element, the noisy position.

The discretization of the continuous time model [8]:

$$F = e^{\Phi(\Delta t)} = I + \Phi(\Delta t) + \frac{1}{2!}\Phi^2(\Delta t)^2 + \dots$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}(\Delta t) + \frac{1}{2}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}^2 (\Delta t)^2 + \dots$$
$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}(\Delta t) + \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}(\Delta t)^2 + \dots$$
$$= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

leads to the discrete time model

$$x(k+1) = Fx(k) + w(k) \Rightarrow \begin{bmatrix} s(k+1) \\ \upsilon(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} s(k) \\ \upsilon(k) \end{bmatrix} + w(k)$$

$$z(k) = Hx(k) + v(k) \Rightarrow z(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} s(k) \\ \upsilon(k) \end{bmatrix} + v(k)$$

Then, the discrete time model parameters are:

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$Q = \begin{bmatrix} \frac{1}{3}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{2}(\Delta t)^2 & \Delta t \end{bmatrix}\sigma_q^2 \qquad (5)$$
$$R = \sigma_r^2$$

The model describes the movement in one axis (x-axis or y-axis). The state vector is of dimension n=2 and contains the position and the velocity: $x(k) = \begin{bmatrix} s(k) & \upsilon(k) \end{bmatrix}^T$. The measurement vector is of dimension m=1 and contains the measured noisy position z(k).

We are able to describe the movement in two axis using two separate state vectors: $x_x(k) = \begin{bmatrix} s_x(k) & \upsilon_x(k) \end{bmatrix}^T$ for the x-axis and $x_y(k) = \begin{bmatrix} s_y(k) & \upsilon_y(k) \end{bmatrix}^T$ for the y-axis.

2.2. Constant acceleration model:

The constant acceleration model describes the constant acceleration movement in two dimensions separately. We start with the constant acceleration movement in one dimension.

Continuous Model:

The acceleration is constant. The state vector has three elements, the position, the velocity and the acceleration and the measurement vector has one element, the noisy position.

$$\dot{x}(t) = \Phi x(t) + Gw(t) \Rightarrow \begin{bmatrix} \dot{s}(t) \\ \dot{\upsilon}(t) \\ \dot{\alpha}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} s(t) \\ \upsilon(t) \\ \alpha(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t)$$

$$z(t) = \mathrm{H}x(t) + v(t) \Rightarrow z(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} s(t) \\ \upsilon(t) \\ \alpha(t) \end{bmatrix} + v(t)$$

Discrete Model:

The velocity is constant. The state vector has three elements, the position, the velocity and the acceleration and the measurement vector has one element, the noisy position.

The discretization of the continuous time model [8]:

$$F = e^{\Phi(\Delta t)} = I + \Phi(\Delta t) + \frac{1}{2!}\Phi^2(\Delta t)^2 + \frac{1}{3!}\Phi^3(\Delta t)^3 + \ldots$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}(\Delta t) + \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}^2 (\Delta t)^2 + \frac{1}{6}\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}^3 (\Delta t)^3 + \ldots$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}(\Delta t) + \frac{1}{2}\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}(\Delta t)^2 + \frac{1}{6}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}(\Delta t)^3 + \ldots$$

$$= \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

leads to the discrete time model

$$x(k+1) = Fx(k) + w(k) \Rightarrow \begin{bmatrix} s(k+1) \\ \upsilon(k+1) \\ \alpha(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} s(k) \\ \upsilon(k) \\ \alpha(k) \end{bmatrix} + w(t)$$

$$z(k) = Hx(k) + v(k) \Rightarrow z(k) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} s(k) \\ \upsilon(k) \\ \alpha(k) \end{bmatrix} + v(k)$$

Then, the discrete time model parameters are:

$$F = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$Q = \begin{bmatrix} \frac{1}{20}(\Delta t)^5 & \frac{1}{8}(\Delta t)^4 & \frac{1}{3}(\Delta t)^3 \\ \frac{1}{8}(\Delta t)^4 & \frac{1}{3}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{6}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 & \Delta t \end{bmatrix}\sigma_q^2 \qquad (6)$$

$$R = \sigma_r^2$$

The model describes the movement in one axis (x-axis or y-axis). The state vector is of dimension n=3 and contains the position, the velocity and the acceleration: $x(k) = \begin{bmatrix} s(k) & \upsilon(k) & \alpha(k) \end{bmatrix}^T$. The measurement vector is of dimension m=1 and contains the measured noisy position z(k).

We are able to describe the movement in two axis using two separate state vectors:

$x_x(k) = \begin{bmatrix} s_x(k) & \upsilon_x(k) & \alpha_x(k) \end{bmatrix}^T$ for the x-axis and $x_y(k) = \begin{bmatrix} s_y(k) & \upsilon_y(k) & \alpha_y(k) \end{bmatrix}^T$ for the y-axis.

3. POSITION ESTIMATION USING STEADY STATE KALMAN FILTER

Concerning the position estimation the aim is to estimate the state variable, using the Kalman filter [9], which is the most well-known algorithm that solve the estimation problem by computing the estimate value x(k/k) of the state vector and the corresponding estimation error covariance matrix P(k/k) at time k. For time invariant systems, where the system transition matrix, the output matrix, the plant and measurement noise covariance matrices are constant, the resulting time invariant Kalman filter takes the following form:

Time Invariant Kalman filter:

$$\begin{aligned} x(k+1/k) &= Fx(k/k) \\ P(k+1/k) &= Q + FP(k/k)F^T \\ K(k+1) &= P(k+1/k)H^T\left[HP(k+1/k)H^T + R\right]^{-1} \\ x(k+1/k+1) &= \left(I - K(k+1)H\right)x(k+1/k) + K(k+1)z(k+1) \\ P(k+1/k+1) &= \left(I - K(k+1)H\right)P(k+1/k) \end{aligned} \quad (7)$$

For k=0,1,… , with initial conditions $x(0/0)=x_0, P(0/0)=P_0$.

For time invariant systems, it is well known [9] that if pair [F H^T] is completely detectable and the pair [F^T Q] is completely stabilizable, then the prediction estimation error covariance matrix tends to a constant value as time tends to infinity. Thus there exists a unique steady state value P_p of the prediction error covariance matrix; there also exists a unique steady state value P_e of the estimation error covariance matrix and a unique steady state value K of the Kalman filter gain. These values remain constant after the steady state time is reached. In this case, the resulting discrete time steady state Kalman filter takes the following form:

Steady state Kalman filter:

$$x(k+1/k+1) = Ax(k/k) + Bz(k+1) \quad (8)$$

for k=0,1,…

with initial condition $x(0/0)=x_0$

where the matrices

$$A = \left(I - KH\right)F \quad \text{and} \quad B = K \quad (9)$$

are calculated off-line by first solving the corresponding discrete time Riccati equation [9]-[14]:

$$P_p = Q + FP_pF^T - FP_pH^T\left[HP_pH^T + R\right]^{-1}HP_pF^T \quad (10)$$

and then calculating the steady state Kalman filter gain

$$K = P_pH^T[HP_pH^T + R]^{-1} \quad (11)$$

The steady state estimation error covariance is

$$P_e = \left(I - KH\right)P_p = P_p - P_pH^T\left[HP_pH^T + R\right]^{-1}HP_p \quad (12)$$

4. POSITION PREDICTION USING STEADY STATE KALMAN FILTER

Concerning the position prediction, the aim is to predict the state variable N time units ahead of the present time (N≥1), i.e. to compute the prediction x(k+N/k) of the state vector at time k+N, given the measurements till time k.

The one step prediction (N=1) is given by the Kalman filter equations [9]:

$$x(k+1/k) = Fx(k/k) \quad (13)$$

It is clear that the computation of the one step prediction requires the knowledge of the estimation x(k/k) of the state vector at time k, which can be derived using the steady state Kalman filter.

The multiple steps, N, (N≥2), prediction is given by the equations [9]:

$$x(k+N/k) = F^{N-1}x(k+1/k) \tag{14}$$

It is clear that the computation of the prediction requires the knowledge of the estimation, which can be derived using the steady state Kalman filter.

One step prediction:

In the steady state case, from (8) and (13), we take:

$$x(k+1/k) = Fx(k/k) = F\left[\left(I-KH\right)x(k/k-1)+Kz(k)\right] = F\left(I-KH\right)x(k/k-1)+FKz(k)$$

The resulting steady state one step prediction algorithm takes the following form:

$$x(k+1/k) = A_1 x(k/k-1) + B_1 z(k) \tag{15}$$

for k=1,2,…, with initial condition $x(1/0) = Fx(0/0) = Fx_0$,

where

$$A_1 = FAF^{-1} = F\left(I-KH\right) \quad \text{and} \quad B_1 = FB = FK \tag{16}$$

Two steps prediction:

From the time invariant two step prediction algorithm, we take:

$$\begin{aligned} x(k+2/k) = Fx(k+1/k) &= F\left[A_1 x(k/k-1) + B_1 z(k)\right] \\ &= FA_1F^{-1}Fx(k/k-1) + FB_1 z(k) \\ &= FA_1F^{-1}x(k+1/k-1) + FB_1 z(k) \end{aligned}$$

The resulting steady state two steps prediction algorithm takes the following form:

$$x(k+2/k) = A_2 x(k+1/k-1) + B_2 z(k) \tag{17}$$

for k=1,2,…, with initial condition $x(2/0) = Fx(1/0) = F^2x(0/0) = F^2x_0$,

where

$$A_2 = F^2AF^{-2} = F^2\left[I-KH\right]F^{-1} \quad \text{and} \quad B_2 = F^2B = F^2K. \tag{18}$$

Multiple steps prediction:

The resulting steady state multiple steps prediction algorithm takes the following form:

$$x(k+N/k) = A_N x(k+N-1/k-1) + B_N z(k) \tag{19}$$

for k=1,2,… and N≥2, with initial condition $x(N/0) = F^N x(0/0) = F^N x_0$,

where

$$A_N = F^N AF^{-N} = F^N\left[I-KH\right]F^{-(N-1)} \quad \text{and} \quad B_N = F^N B = F^N K. \tag{20}$$

It is obvious that the matrices in (19) are calculated off-line by first solving the corresponding discrete time Riccati equation (10) and then calculating the steady state Kalman filter gain using (11).

5. SIMULATION

5.1 Constant velocity model:

The use of the constant velocity model, which describes the movement in x-axis and y-axis separately, requires the use of two steady state Kalman filters in order to compute the position estimation for each movement. Similarly, the position prediction requires the use of two steady state predictions algorithms. The two estimation algorithms as well the two prediction algorithms can be implemented in parallel.

Simulation parameters:

Velocity: $\upsilon_x(t) = 20, \upsilon_y(t) = 30$.

Model parameters:

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$
$$Q = \begin{bmatrix} \frac{1}{3}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{2}(\Delta t)^2 & \Delta t \end{bmatrix} \sigma_q^2, \qquad \sigma_q^2 = 0.01$$
$$R = \sigma_r^2, \qquad \sigma_r^2 = 0.1$$

Discretization time Δt=1 and initial condition x(0/0)=0.

Estimation:

The position estimation using steady state Kalman filter for the constant velocity model is depicted in Figure 1. The real position and the steady state Kalman filter estimation are plotted.

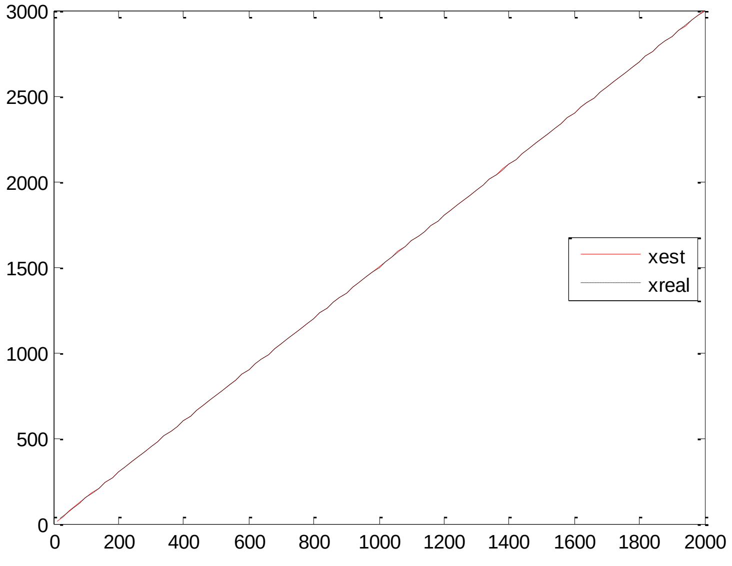

Fig 1. Position estimation for constant velocity movement

Simulation was performed:

- for various values of the discretization time

- for various values of the measurements noise covariance concerning the trust of the measurements

- for various velocities Comments.

The steady state Kalman filter produces satisfying estimates.

The discretization time does not affect the filter.

The trust of the measurements affects the filters: the filters produce worse estimates as measurements noise covariance increases.

The movement's velocity does not affect the filter.

Prediction:

The position prediction using steady state prediction algorithms for the constant velocity model is depicted in Figure 2. The real position and the one step prediction are plotted.

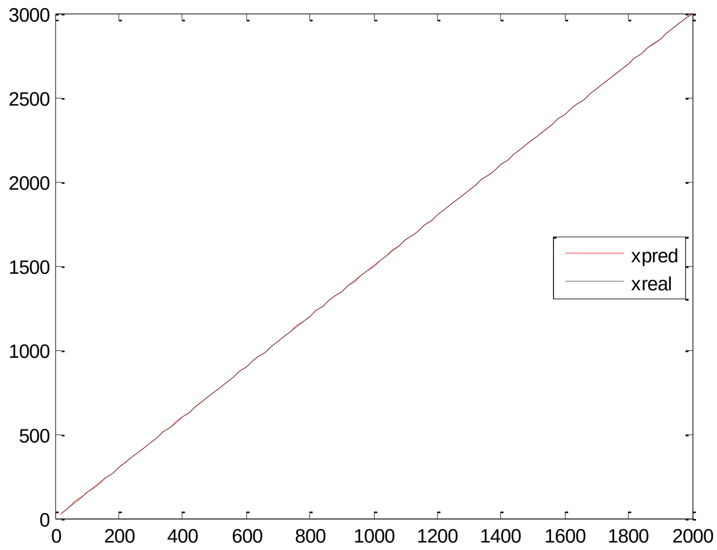

Fig.2. Position prediction for constant velocity movement

Simulation was performed:

- for various values of the steps prediction ahead of the present time

- for various values of the discretization time

- for various values of the measurements noise covariance concerning the trust of the measurements

- for various velocities

Comments.

The steady state prediction algorithms for small time units ahead of the present time produce satisfying predictions compared to one step prediction.

The discretization time does not affect the prediction algorithms.

The trust of the measurements affects the prediction algorithms: the prediction algorithms produce worse predictions as measurements noise covariance increases.

The movement's velocity does not affect the prediction algorithms.

5.2 Constant acceleration model:

The use of the constant acceleration model, which describes the movement in x-axis and y-axis separately, requires the use of two steady state Kalman filters in order to compute the position estimation for each movement. Similarly, the position prediction requires the use of two steady state predictions algorithms. The two estimation algorithms as well the two prediction algorithms can be implemented in parallel.

Simulation parameters:

Acceleration: $\alpha_x(t) = 2, \alpha_y(t) = 3$.

Model parameters:

$$F = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

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

$$Q = \begin{bmatrix} \frac{1}{20}(\Delta t)^5 & \frac{1}{8}(\Delta t)^4 & \frac{1}{6}(\Delta t)^3 \\ \frac{1}{8}(\Delta t)^4 & \frac{1}{3}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 \\ \frac{1}{6}(\Delta t)^3 & \frac{1}{2}(\Delta t)^2 & \Delta t \end{bmatrix}\sigma_q^2, \quad \sigma_q^2 = 0.01$$

$$R = \sigma_r^2, \sigma_r^2 = 0.1$$

Discretization time Δt=1 and initial condition x(0/0)=0.

Estimation:

The position estimation using steady state Kalman filter for the constant acceleration model is depicted in Figure 3. The real position and the steady state Kalman filter estimation are plotted.

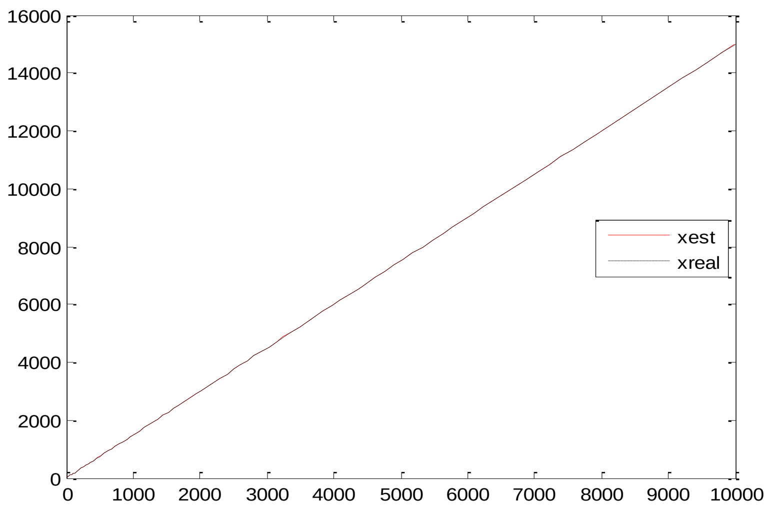

Fig.3. Position estimation for constant acceleration movement

Simulation was performed:

- for various values of the discretization time

- for various values of the measurements noise covariance concerning the trust of the measurements

- for various velocities Comments.

The steady state Kalman filter produces satisfying estimates.

The discretization time does not affect the filter.

The trust of the measurements affects the filters: the filters produce worse estimates as measurements noise covariance increases.

The movement's velocity does not affect the filter.

Prediction:

The position prediction using steady state prediction algorithms for the constant acceleration model is depicted in Figure 4. The real position and the one step prediction are plotted.

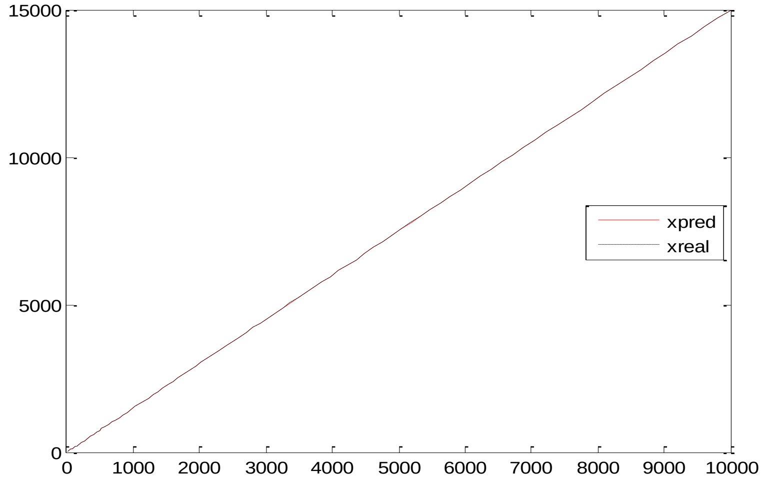

Fig. 4. Position prediction for constant acceleration movement

Simulation was performed:

- for various values of the steps prediction ahead of the present time

- for various values of the discretization time

- for various values of the measurements noise covariance concerning the trust of the measurements

- for various velocities Comments.

The steady state prediction algorithms for small time units ahead of the present time produce satisfying predictions compared to one step prediction.

The discretization time does not affect the prediction algorithms.

The trust of the measurements affects the prediction algorithms: the prediction algorithms produce worse predictions as measurements noise covariance increases.

The movement's velocity does not affect the prediction algorithms.

5.3 Position estimation and prediction absolute average error:

In order to understand the behavior of the steady state estimation and prediction algorithms, the percent absolute average error for the two models are presented in Table 1.

Table 1. Percent absolute average error of position estimation and prediction

estimation		
steady state Kalman filter	constant velocity	constant acceleration
	0.0111	0.0077
Prediction		
prediction steps	constant velocity	constant acceleration
1	0.0142	0.0146
2	0.0156	0.0173
3	0.0165	0.0189
4	0.0172	0.0200
5	0.0177	0.0209

5.4 Position estimation and prediction calculation burden:

The steady state Kalman filter and the steady state prediction algorithms are iterative algorithms. Thus, we are interested in calculating their per iteration calculation burden required for the on-line calculations; the calculation burden of the off-line calculations (initialization process) is not taken into account. The coefficients of the steady state Kalman filter in (9) are calculated off-line. The coefficients of the steady state prediction algorithms in and (20) are also calculated off-line. Table 2 summarizes the per-iteration calculation burden of the steady state Kalman filter and the steady state prediction algorithms, computed using the ideas in [15].

Table 2. Per-iteration calculation burden of position estimation and prediction algorithms

estimation		
matrix operation	dimensions	calculation burden
$Ax(k/k)$	$(n\times n)*(x\times 1)$	$2n^2-n$
$Bz(k+1)$	$(n\times m)*(m\times 1)$	$2nm-n$
$x(k+1/k+1)=Ax(k/k)+Bz(k+1)$	$(n\times 1)+(n\times 1)$	n
total calculation burden		$2n^2+2nm-n$
prediction		
matrix operation	dimensions	calculation burden
$A_N x(k/k-1)$	$(n\times n)*(x\times 1)$	$2n^2-n$
$B_N z(k+1)$	$(n\times m)*(m\times 1)$	$2nm-n$
$x(k+N/k)=A_N x(k/k-1)+B_N z(k)$	$(n\times 1)+(n\times 1)$	n
total calculation burden		$2n^2+2nm-n$

Note that the per-iteration calculation burden of all prediction algorithms is constant since it is independent of the prediction steps. Note also that the per-iteration calculation burden of estimation equals the per-iteration calculation burden of prediction.

It is obvious that the per-iteration calculation burden of the estimation and prediction algorithms depends on the state vector dimension n and the measurement vector dimension m. The matrix operations involved in steady state estimation/prediction algorithms are matrix addition and matrix multiplication; thus the steady state estimation/prediction has complexity $O\left(\max(n^2,nm)\right)$. On the other hand, matrix inversion is involved in the Kalman filter equations (7); thus the Kalman filter has complexity $O\left(\max(n^3,m^3)\right)$ [15]. Hence, the derived steady state estimation/prediction algorithms are much faster than the Kalman filter.

In the constant velocity model, the state vector is of dimension n=2 and the measurement vector is of dimension m=1. In the constant acceleration model, the state vector is of dimension n=3 and the measurement vector is of dimension m=1. Thus the per-iteration calculation burden of position estimation/prediction is $CB_v=10$ for the constant velocity model and $CB_a=21$ for the constant acceleration model. In both models the estimation/prediction can be derived using *two* estimation/prediction algorithms sequentially or in parallel, one for the x-axis and the other for the y-axis.

6. CONCLUSIONS

Two models for Mobile Position Tracking (MPT), which describe constant velocity and constant acceleration movements in two dimensions, were presented. Steady state estimation and prediction algorithms based on the steady state Kalman filter were derived. Simulation results show that the position estimation and prediction are reliable. It was shown that the estimation calculation burden equals the prediction calculation burden; this steady state estimation/prediction calculation is much less than the calculation burden of the Kalman filter.

Thus Mobile Position Tracking (MPT) can be realized for position estimation or prediction using using two steady state estimation/prediction algorithms sequentially or in parallel, one for the x-axis and the other for the y-axis. The estimation/prediction algorithms compute estimates that are very close to the real position and are faster than classical Kalman filter.

REFERENCES

[1] Drane C, Macnaughtan M, Scott C. Positioning GSM telephones. IEEE Communications Magazine 1998; 36(4): 46-54.

[2] Küpper A. Location-based services. Fundamentals and Operation. John Wiley & Sons 2005.

[3] Mouly M, Pautet M. The GSM system for mobile communications. Washington DC: Telecom Publishing 1992.

[4] Rooney S, Chippendale P, Choony R, Le Roux C, Honary B. Accurate vehicular positioning using DAB-GSM hybrid system, VTC 2000-Spring Tokyo conference proceedings, IEEE 51st, 2000; 1: 97-101.

[5] Dubois JP, Daba JS, Nader M, El Ferkh C. GSM Position Tracking using a Kalman Filter. World Academy of Science, Engineering and Technology 2012; 68: 1610-19.

[6] Assimakis N, Adam M. Global systems for mobile position tracking using Kalman and Lainiotis filters. The Scientific World Journal 2014; 2014, Article ID 130512: 1-8, http://dx.doi.org/10.1155/2014/130512.

[7] Assimakis N., Adam M., "Mobile Position Tracking in Three Dimensions using Kalman and Lainiotis Filters", The Open Mathematics Journal, vol. 8, pp. 1-6, 2015.

[8] Ian Reid, "Estimation II", 2001.http://www.robots.ox.ac.uk/~ian/Teaching/Estimation/LectureNotes2.pdf

[9] Anderson BDO, Moore JB. Optimal Filtering. New York: Dover Publications 2005.

[10] Assimakis N, Lainiotis D, Katsikas S, Sanida F. A survey of recursive algorithms for the solution of the discrete time Riccati equation. Nonlinear Analysis, Theory, Methods & Applications 1997; 30(4): 2409-20.

[11] Assimakis N, Roulis S, Lainiotis D. Recursive solutions of the discrete time Riccati equation. Neural, Parallel and Scientific Computations 2003; 11: 343-50.

[12] Lainiotis DG, Assimakis ND, Katsikas SK. A new computationally effective algorithm for solving the discrete Riccati equation. Journal of Mathematical Analysis and Applications 1994; 186(3): 868-95.

[13] Lainiotis DG, Assimakis ND, Katsikas SK. 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 1995; 3(4): 565-84.

[14] Vaughan DR. A nonrecursive algebraic solution for the discrete time Riccati equation. IEEE Transactions on Automatic Control 1970; 15(5): 597-9.

[15] Assimakis N, Adam M. Discrete time Kalman and Lainiotis filters comparison. Int. Journal of Mathematical Analysis (IJMA) 2007; 1(13): 635-59.