Optimal linear interpolation algorithm with fixed delay

Anastasia Cherkasova¹* and Alexander Shatilov¹

¹National Research University MPEI, Moscow, Russian Federation

Abstract. One of the most important characteristics of satellite navigation receiving equipment is the accuracy of navigation definitions. To improve this characteristic, the paper considers the possibility of using optimal interpolation algorithms with a fixed delay. A new interpolation algorithm with a fixed delay in discrete time was deduced. The use of the proposed interpolation algorithm instead of filtering made it possible to reduce the standard deviation by 2 times with a delay in the output of the result by 10 cycles. The advantage of the proposed algorithm is the reduction in computational complexity compared to the optimal two-way algorithm. The proposed algorithm requires 2 matrix accesses per cycle of operation, subject to the use of additional memory resources, to save pre-calculated elements.

1 Introduction

In the theory of optimal estimation, three general tasks of estimating random processes are distinguished separately: interpolation, filtration and extrapolation [1, p. 674]. These tasks differ in the combinations between the time of the last observation and the time of the formation of the process estimation. In the filtration task, the estimation of the process is formed synchronously with the incoming observations. An extrapolation estimate is generated for some point in the future relative to the last received observation. If an estimate is formed for a certain moment in the past, relative to the last observation, then the estimation task is called interpolation. In navigation systems, it is filtration estimates that are most widely used. This is due to the need to provide the consumer with real-time navigation parameters.

This is due to the need to provide the consumer with real-time navigation parameters. However, the accuracy of the interpolation estimate should be higher than that of the filtration estimate, since additional observations from future time points relative to the time of the estimate can be used to form the interpolation estimate. The disadvantage of interpolation estimates is the delay relative to the current time. However, in some cases such a delay may be acceptable. Often in practice, due to computational and protocol delays, even filtration estimates are issued to the consumer with a significant delay.

* Corresponding author: PotrikeevaAA@mpei.ru
The interpolation estimate of the process at time \( t_v \) denoted as \( \mathbf{X}_v \). The arrival time of the last observation is denoted as \( t_k \). Then the delay of the interpolation estimate:

- \( \tau = t_k - t_v \) in seconds;
- \( \delta = k - v \) in samples.

In the theory of optimal estimation, there are three variants of interpolation [1, p. 678]:

- Interpolation at a fixed point. A moment of time \( t_v \) is fixed. Need to get an estimate of the process \( \mathbf{X}_v \) at a moment in time \( t_v \) (\( \tau \) increases with the number of observations).
- Interpolation at a fixed interval. The estimation of the process \( \mathbf{X}_v \) is carried out at any point in time within a fixed interval \([0, \tau)\);
- Interpolation with a fixed delay. Process estimation is performed with a fixed delay relative to the last observation, i.e. \( \tau = t_k - t_v = \text{const} \).

Next, fixed-delay interpolation is considered, since this type of interpolation can act as an alternative to filtration.

Algorithms for linear interpolation with a fixed delay are widely used in radar to improve the accuracy of tracking highly maneuverable targets [2-11]. The two-way interpolation algorithm presented in [2-5] was derived from filtration algorithms and showed a good result. The disadvantage of this algorithm is the high computational load. One cycle of operation the two-way interpolation algorithm requires \( \delta + 3 \) matrix inversion. In the article [12] the article presents an algorithm for optimal linear interpolation with a fixed delay, derived by approximately finding a recursive equation for the posterior probability density (PPD) of an interpolation estimate, similar to the Stratonovich equation for a filtration estimate in discrete time. However, during simulation, this algorithm showed instability with an increase in the fixed delay time. Algorithm [16] due to instability, led to divergent results when trying to implement it in practice.

Following is the output of the linear interpolation algorithm.

2 Formulation of the task

In discrete time \( t_k \), where is \( k = 1,2,... \) the process implementation is observed:

\[
y_k = \lambda_k + n_k
\]

Where is \( y \) – the vector of observations of size \( M \), \( n_k \) – discrete white Gaussian noises (DWGN) vector with zero mean and correlation matrix \( D_n \), \( \lambda \) - vector of parameters to be estimated.

Let \( \lambda_k \) can be represented as separate components of a vector Markov sequence \( x_k \):

\[
\lambda_k = H \cdot x_k
\]

\( H \) – known matrix, \( x \) – state vector of size \( N \).

We define the model of the vector Markov sequence by the linear difference equation:

\[
x_k = F \cdot x_{k-1} + G \cdot \xi_{k-1}
\]

where is \( x \) – state vector of size \( N \), \( k \) – current sample number in the sequence; \( \xi \) – vector of discrete DWGN with zero mean and correlation matrix \( D_\xi \), \( F \), \( G \) – known matrices.

It is required to form a Bayesian estimate of the process according to the criterion of the minimum error variance at the moment of time:
\[ \nu = k - \delta \]

where is \( \delta \) – the amount of delay that is considered given.

The Bayesian estimate under the chosen minimum error variance criterion can be found as the first moment \( \text{PPD} \, \hat{x}_v = \int_{x_v} x_v \cdot p(x_v | Y_1^k) \, dx_v \)

\[ (5) \]

3 Deduced of equations for the posterior probability density

To solve this task, it is necessary to find an equation for the posterior probability density (PPD): \( p(x_v | Y_1^k) \).

For this, consider PPD \( p(X_1^k | Y_1^k) \), where is:
- \( Y_1^k = [y_1, ..., y_k] \) - sample of independent observations;
- \( X_1^k = [x_1, ..., x_k] \) - vector Markov sequence.

According to the Bayes' theorem:

\[
p(X_1^k | Y_1^k) = \frac{p(Y_1^k | X_1^k) \cdot p(X_1^k)}{p(Y_1^k)}
\]

Because \( p(Y_1^k) \) – does not depend on \( X_1^k \), introduce a normalization constant \( c_1 = 1/p(Y_1^k) \), hence:

\[
p(X_1^k | Y_1^k) = c_1 \cdot p(Y_1^k | X_1^k) \cdot p(X_1^k)
\]

\[ (7) \]

By the condition of the task, the readings of the observation noise are independent. Therefore, in accordance with the equations (1)-(2), \( y_j \) depends only on \( x_j \), and does not depend on \( y_2...y_k \) and \( x_2...x_k \). It can also be said that \( y_j \) does not depend on \( x_j \), provided the \( x_2...x_k \) are known. Then

\[
p(Y_1^k | X_1^k) = p(y_1 | x_1) \cdot p(y_2...y_k | x_2...x_k)
\]

Continuing the same reasoning for \( y_2...y_k \) we can conclude that:

\[
p(Y_1^k | X_1^k) = \prod_{j=2}^{k} p(y_j | x_j)
\]

\[ (8) \]

Where is \( p(y_j | x_j) \) – one-step likelihood function.

Given the fact that \( X_1^k \) – is a vector Markov sequence:

\[
p(X_1^k) = p(x_k | x_1, x_2, ..., x_{k-1}) \cdot p(x_1, x_2, ..., x_{k-1}) =
= p(x_k | x_{k-1}) \cdot p(x_1, x_2, ..., x_{k-1}) =
= p(x_k | x_{k-1}) \cdot p(x_{k-1}, x_2, ..., x_{k-2}) \cdot p(x_1, x_2, ..., x_{k-2}) =
= p(x_k | x_{k-1}) \cdot p(x_{k-1}, x_{k-2}) \cdot p(x_1, x_2, ..., x_{k-2}) = ...
\]

\[ (9) \]

Continuing similar calculations until \( x_k \), we get:

\[
p(X_1^k) = \prod_{j=1}^{k} p(x_j | x_{j-1})
\]

\[ (10) \]

Where \( p(x_j | x_{j-1}) \) – probability density (PD) of the transition.
At the initial moment of time \( p(x_1|x_0) = p(x_1) \), where is \( p(x_1) \) - prior probability density.

Express the PPD \( p(x_k^k|y_k^k) \) taking into account the equations (7),(8),(10):

\[
p(x_k^k|y_k^k) = c_1 \prod_{j=1}^{k} p(y_j|x_j) \cdot p(x_j|x_{j-1})
\]

(11)

The one-step likelihood function on the right side \( p(y_j|x_j) \) is actually given by the observational model (1)-(2). The transition probability density of a Markov process \( p(x_j|x_{j-1}) \) is actually given by the dynamic equation (3). The constant \( c_1 \) can be obtained from the normalization condition \( \int_{-\infty}^{\infty} p(x)dx = 1 \).

From the consistency property of joint PD:

\[
p(x_k|y_1^k) = \prod_{i=1}^{k} \int_{-\infty}^{\infty} p(y_i|x_i) \cdot p(x_i|x_{i-1}) dX_{i-1}^k dx_i \equiv \prod_{i=1}^{k} p(x_i|y_i^i) dX_{i-1}^k dx_i
\]

(12)

Substituting (11) into (12) we get:

\[
p(x_k|y_1^k) = c_1 \prod_{i=1}^{k} \int_{-\infty}^{\infty} p(y_i|x_i) \cdot p(x_i|x_{i-1}) dX_{i-1}^k dx_i = c_1 \prod_{i=1}^{k} \int_{-\infty}^{\infty} p(x_i|y_i^i) dX_{i-1}^k dx_i
\]

(13)

**Derivation of recursive equations for the PPD of the interpolation estimate.** Denote \( \tilde{p}(x_i|y_i^i) \) - as the unnormalized probability density of the value \( x_i \) provided that \( y_1^i...y_k \) observations are known:

\[
\tilde{p}(x_i|y_i^i) = \prod_{j=1}^{i} \int_{-\infty}^{\infty} p(y_i|x_i) \cdot p(x_i|x_{i-1}) dX_{i-1}^k dx_i
\]

(14)

The unnormalized probability density of the value \( x_i \) provided that the \( y_{i+1}^i...y_k \) observations are known will be denoted as \( \tilde{p}(x_i|y_{i+1}^k) \):

\[
\tilde{p}(x_i|y_{i+1}^k) = \prod_{j=1}^{i} \int_{-\infty}^{\infty} p(y_i|x_i) \cdot p(x_i|x_{i-1}) dX_{i-1}^k dx_i
\]

(15)

Then taking into account the equations (13)-(15):

\[
p(x_i|y_1^k) = c_1 \cdot \tilde{p}(x_i|y_1^k) \cdot \tilde{p}(x_i|y_{i+1}^k)
\]

(16)

It can be shown that \( \tilde{p}(x_i|y_i^i) \) corresponds to the PPD for the filtration task, provided that the last observation obtained is \( y_i \). Recursive equation for finding \( \tilde{p}(x_i|y_i^i) \), similar to the Stratonovich equation for the evolution of PPD:

\[
\tilde{p}(x_i|y_i^i) - c_1 \cdot p(y_i|x_i) \int_{-\infty}^{\infty} p(x_i|x_{i-1}) \tilde{p}(x_i|y_{i-1}^i) dx_{i-1}
\]

(17)
Now consider \( p(x_i | y_{v+1}^1) \). The value of the unnormalized probability density, provided that \( y_{v+1} \ldots y_k \) observations are known can be represented as:

\[
\tilde{p}(x_i | y_{v+1}^1) = \int \prod_{j=v+1}^k p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_{v+1} = \]

\[
= \int p(y_v | x_i) \int p(x_i | x_{v-1}) \cdot \prod_{j=v+1}^{k-1} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_i dx_{v+1} = 
\]

\[
= \int p(y_v | x_i) \int p(x_i | x_{v-1}) \cdot p(y_{v-1} | x_{v-1}) \times 
\]

\[
\times \int p(x_{v-1} | x_{v-2}) \cdot \prod_{j=v+1}^{k-2} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_i dx_{v-1} dx_{v+1} dx_{v-1}^{k-2} 
\]

Consider separately the multiple integral over \( dx_{v+1}^{k-2} \):

\[
\int p(x_{v-1} | x_{v-2}) \cdot \prod_{j=v+1}^{k-2} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_{v+1}^{k-2} = 
\]

\[
= \int p(x_{v-1} | x_{v-2}) \cdot p(y_{v-2} | x_{v-2}) \times 
\]

\[
\times \int p(x_{v-2} | x_{v-3}) \cdot \prod_{j=v+1}^{k-3} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_i dx_{v-2} dx_{v-1}^{k-3} 
\]

Let us denote the expression under the integral over \( dx_{v+1}^{k-2} \) as the unnormalized conditional PD:

\[
\tilde{p}(x_{v-1} | y_{v+1}^{k-2}, x_i) = \int p(x_{v-1} | x_{v-2}) \cdot \prod_{j=v+1}^{k-2} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_{v+1}^{k-2} 
\]

Then, one of the factors on the right side can be denoted as:

\[
\tilde{p}(x_{v-2} | y_{v+1}^{k-3}, x_i) = \int p(x_{v-2} | x_{v-3}) \cdot \prod_{j=v+1}^{k-3} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_{v-2} dx_{v-1}^{k-3} 
\]

It is possible to write down the general formula for the non-normalized conditional PD:

For \( i \in [2, k-v] \):

\[
\tilde{p}(x_{v+i-1} | y_{v+1}^i, x_i) = \int p(x_{v+i} | x_{v+i-1}) \cdot \prod_{j=v+1}^{i-1} p(y_j | x_i) \cdot p(x_i | x_{j-1}) dx_{v+i-1}^{i-1} 
\]

Then for a step \( i = 2 \):

\[
\tilde{p}(x_{v+1}^2 | y_{v+1}^1, x_i) = \int p(x_{v+2} | x_{v+1}) \cdot p(y_{v+1} | x_{v+1}) \cdot p(x_{v+1} | x_i) dx_{v+1} 
\]

For a step \( i = (k-v) \):

\[
\tilde{p}(x_k | y_{v+1}^{k-2}, x_i) = \int p(x_k | x_{k-1}) \cdot \tilde{p}(y_{k-1} | x_{k-1}) \cdot \tilde{p}(x_{k-1} | y_{v+1}^{k-3}, x_i) dx_{k-1} 
\]

\[
\tilde{p}(x_i | y_{v+1}^k) = \int p(y_v | x_i) \cdot \tilde{p}(x_i | y_{v+1}^{k-1}, x_i) dx_i 
\]
The equations (23)-(24) allow us to recursively find the function \( \tilde{p}(x_i | Y_{v+1}^{i-1}, x_r) \) which at the last step gives the desired probability density \( \tilde{p}(x_r | Y_{v+1}^k) \).

As a result, we obtain an algorithm for finding PPD \( p(x_v | Y_v^k) \):

\[
\tilde{p}(x_v | Y_v^k) = c_1 \cdot p(y_v | x_v) \int p(x_v | x_{v-1}) \tilde{p}(x_{v-1} | Y_{v+1}^{i-1}) dx_{v-1}
\]

\[
\tilde{p}(x_v | Y_v^k) = \sum_{i \in [2,k-v]} \int p(y_v | x_v) \tilde{p}(x_v | Y_v^{i-1}, x_r) dx_r
\]

(25)

\[
p(x_v | Y_v^k) = c_1 \cdot \tilde{p}(x_v | Y_v^k) \cdot \tilde{p}(x_v | Y_v^{i-1})
\]

4 Derivation of recurrent equations for \( \tilde{p}(x_v | Y_v^k) \) with Gaussian approximation of PPD

Let’s pass from the recurrence relation for the PPD \( \tilde{p}(x_v | Y_v^k) \) to the recurrence relations for the moments of this PPD (mean and variance). Let us write expressions for the probability densities included in the non-normalized conditional PD \( \tilde{p}(x_v | Y_v^{i-1}, x_r) \) for \( i \in [2,k-v] \) taking into account the Gaussian distribution due to the observation noise.

\[
p(x_v | x_{v-1}) = \frac{1}{(2\pi)^{N/2} \sqrt{\det(D_z)}} \exp\left\{ -\frac{1}{2} (x_v - Fx_{v-1})^T D_z^{-1} (x_v - Fx_{v-1}) \right\}
\]

\[
p(y_{v+1} | x_{v+1}) = \frac{1}{(2\pi)^{M/2} \sqrt{\det(D_n)}} \exp\left\{ -\frac{1}{2} (y_{v+1} - Hx_{v+1})^T D_n^{-1} (y_{v+1} - Hx_{v+1}) \right\}
\]

(26)

Where is: \( N \) – state vector \( x \) dimension, \( M \) – observation vector \( y \) dimension.

\[
\int \exp\left\{ -x'Ax + b'x + c \right\} dx = \sqrt{\frac{\pi}{\det(A)}} \exp\left\{ \frac{1}{4} b' A^{-1} b + c \right\}
\]

(27)

And using the equations (26)-(27) we derive a general formula for the non-normalized conditional PD \( \tilde{p}(x_v | Y_v^{i-1}, x_r) \) for \( i \in [2,k-v] \) taking into account the Gaussian approximation:
\[ \tilde{p}(x_{i\mid v+1}, x_v) = C_i \cdot \exp \left\{ -\frac{1}{2} \left[ x_{i\mid v+1} D_{i\mid v+1}^{-1} x_{v+1} + S_{v+1} + R_{v+1} \right] \right\} \times \]
\[ \times \exp \left\{ -\frac{1}{2} \left[ \left( F' D_{i\mid v+1}^{-1} x_{v+1} + M_{v+1} \right)^T \cdot D_{v+1}^{-1} \cdot \left( F' D_{i\mid v+1}^{-1} x_{v+1} + M_{v+1} \right) \right] \right\} \]
\[ C_i = \left( \frac{2\pi}{\left(\frac{(i-1)!}{(j-1)!} M_{i\mid v}^{(j)}\right)^{i-j}} \right) \cdot \frac{2 \cdot \det(D_i) \cdot \det(D_v)}{\sqrt{\pi^j \prod_{j=2}^i \det(D_{v+1})}} \]

(28)

The PD components \( D_{v+1}, M_{v+1}, S_{v+1}, R_{v+1} \) are found iteratively as:

for \( i = 2 \); \( D_{v+2} = H^T D_n^{-1} H + D_z^{-1} + F^T D_z^{-1} F \)

for \( i = [3, k-v] \);

\( D_{v+2} = D_{v+2} - D_z^{-1} F(D_{v+1}^{-1})^{-1} F^T D_z^{-1} \)

\( M_{v+2} = H^T D_n^{-1} y_{v+1} + D_z^{-1} F x_v \)

\( M_{v+2} = H^T D_n^{-1} y_{v+1} + D_z^{-1} F(D_{v+1}^{-1})^{-1} M_{v+1} \)

\( S_{v+2} = y_v^T D_n^{-1} y_{v+1} \)

\( S_{v+2} = y_v^T + D_n^{-1} y_{v+1} \)

\( R_{v+2} = (F x_v)^T D_z^{-1} F x_v \)

\( R_{v+2} = R_{v+2} - M_{v+1}^{-1} (D_{v+1}^{-1})^{-1} M_{v+1} \) (29)

Now we substitute the general formula for the non-normalized conditional PD \( \tilde{p}(x_{i\mid v+1}, x_v) \) for \( i \in [2, k-v] \) into the formula for the non-normalized probability density \( \tilde{p}(x_i | Y_{v+1}) \) (equations (25)):

For \( i = k-v \);

\[ p(x_i | Y_{v+1}) = \int p(y_k | x_i) \cdot \tilde{p}(x_{i\mid v+1}, x_v) dx \]

(30)

Taking into account the Gaussian distribution of observational noise:

\[ p(y_k | x_i) = \frac{1}{\sqrt{2\pi M_{v+1}^{(i)}}} \exp \left\{ -\frac{1}{2} (y_k - H x_k)^T D_{v+1}^{-1} (y_k - H x_k) \right\} \]

(31)

Substitute (28) and (31) into the equation (30):

\[ p(x_i | Y_{v+1}) = C_i \cdot \int \exp \left\{ -\frac{1}{2} (y_k - H x_k)^T D_{v+1}^{-1} (y_k - H x_k) \right\} \times \]
\[ \times \exp \left\{ \frac{1}{2} \left[ (F' D_{v+1}^{-1} x_{v+1} + M_{v+1})^T \cdot D_{v+1}^{-1} \cdot (F' D_{v+1}^{-1} x_{v+1} + M_{v+1}) \right] \right\} \times \]
\[ \times \exp \left\{ -\frac{1}{2} \left[ x_{i\mid v+1} D_{i\mid v+1}^{-1} x_{v+1} + S_{v+1} + R_{v+1} \right] \right\} dx \]

(32)

We obtain an expression for the unnormalized probability density \( \tilde{p}(x_i | Y_{v+1}) \) taking into account the Gaussian approximation:

\[ \tilde{p}(x_i | Y_{v+1}) = C_i \cdot \exp \left\{ \frac{1}{2} \left[ (H^T D_{v+1}^{-1} y_k + D_z^{-1} F D_{v+1}^{-1} M_{v+1})^T \cdot H^T D_{v+1}^{-1} y_k + D_z^{-1} F D_{v+1}^{-1} M_{v+1} \right] \right\} \]

(33)
The PD components $D_{r+1}, M_{r+1}, S_{r+1}, R_{r+1}$ are found iteratively as:

$$
\text{for } i = 2; \quad D_{r+1} = H^T D_n^{-1} H + D_n^{-1} F D_k^{-1} F^T D_n^{-1} + D_n^{-1} \left(34\right)
$$

$$
\text{for } i = [3, k - 1]; \quad D_{r+1} = D_{r+2} - D_n^{-1} F(D_{r+1})^{-1} F^T D_n^{-1}
$$

$$
M_{r+1} = H^T D_n^{-1} y_{r+1} + D_n^{-1} F x_v
$$

$$
S_{r+1} = y_v^T D_n^{-1} y_{r+1}
$$

$$
R_{r+1} = (F x_v)^T D_n^{-1} F x_v
$$

5 Derivation of the interpolation estimate and its variance

Let us represent the PPD $p(x_v | Y^k_v)$ taking into account the Gaussian distribution of observational noise:

$$
p(x_v | Y^k_v) = \frac{1}{(2\pi)^{M/2} \sqrt{\det(D_{int,v})}} \exp \left\{ \frac{1}{2} (x_v - x_{int,v})^T D_{int,v}^{-1} (x_v - x_{int,v}) \right\} \left(35\right)
$$

Where is $x_{int,v}$ - interpolation estimate, $D_{int,v}$ - interpolation variance.

$$
P = D_{int,v}^{-1}
$$

$$
q^v = x_{int,v}^\top D_{int,v}^{-1}
$$

Then according to the equations (25), (35) and (36):

$$
p(x_v | Y^k_v) = \exp \left\{ \frac{1}{2} (x_v - x_{int,v}^\top P x_v + q^v x_v + c) \right\} = c_s \cdot p(x_v | Y^k_v) \cdot p(x_v | Y^{v+1}) \left(37\right)
$$

Let’s substitute the equation (33) into equation (37):

$$
p(x_v | Y^k_v) = \exp \left\{ \frac{1}{2} (x_v - x_{int,v}^\top P x_v + q^v x_v + c) \right\} = c_s \cdot p(x_v | Y^k_v) \cdot
$$

$$
\exp (\frac{1}{2} \left\{ (H^T D_n^{-1} H - D_n^{-1} F D_k^{-1} F^T D_n^{-1} + D_n^{-1} (y_k^T D_n^{-1} y_k + D_k^{-1} F^T D_k^{-1} M_k) -
$$

$$\cdot y_k^T D_n^{-1} y_k - M_k^T D_k^{-1} M_k + S_k + R_k) \right\}) \left(38\right)
$$

Let us write the expression for the PD $p(x_v | Y^k_v)$ taking into account the Gaussian nature of observational noise and shaping noise:

$$
p(x_v | Y^k_v) = \frac{1}{(2\pi)^{M/2} \sqrt{\det(D_{int,v})}} \exp \left\{ \frac{1}{2} (x_v - \hat{x}_{int,v})^T D_{int,v}^{-1} (x_v - \hat{x}_{int,v}) \right\} \left(39\right)
$$

Where is: $\hat{x}_{int,v}$ - filtration estimate, $D_{int,v}$ - filtration estimate variance.
\[
\exp\left\{-\frac{1}{2} \mathbf{x}_v^T \mathbf{P} \mathbf{x}_v + \mathbf{q}^T \mathbf{x}_v + c\right\} = \\
= \exp\left\{-\frac{1}{2} \mathbf{x}_v^T \left( \mathbf{R}_k^1 + \mathbf{D}_{\text{fr},v}^{-1} - \mathbf{M}_k^T (\mathbf{D}_k)^{-1} \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{W}^{-1} \mathbf{D}_k^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 - \mathbf{M}_k^{1T} \mathbf{D}_k^{-1} \mathbf{M}_k^1\right) \mathbf{x}_v + \\
+ (\mathbf{D}_{\text{fr},v}^{-1})^\top \mathbf{x}_{\text{fr},v} + \mathbf{M}_k^T (\mathbf{D}_k)^{-1} \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{W}^{-1} \mathbf{D}_k^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 - \mathbf{M}_k^{1T} \mathbf{D}_k^{-1} \mathbf{M}_k^1\right) \mathbf{x}_v + \\
+ \frac{1}{2} (\mathbf{U}^\top \mathbf{W}^{-1} \mathbf{U} - \mathbf{x}_{\text{fr},v}^T \mathbf{D}_{\text{fr},v}^{-1} \mathbf{x}_{\text{fr},v} - \mathbf{z}) + \ln(C_0)\right\} \\
\mathbf{P} = \mathbf{R}_k^1 + \mathbf{D}_{\text{fr},v}^{-1} - \mathbf{M}_k^T (\mathbf{D}_k)^{-1} \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{W}^{-1} \mathbf{D}_k^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 - \mathbf{M}_k^{1T} \mathbf{D}_k^{-1} \mathbf{M}_k^1 \\
\mathbf{q}^T = (\mathbf{D}_{\text{fr},v}^{-1})^\top \mathbf{x}_{\text{fr},v} + \mathbf{M}_k^T (\mathbf{D}_k)^{-1} \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{W}^{-1} \mathbf{D}_k^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 - \mathbf{M}_k^{1T} \mathbf{D}_k^{-1} \mathbf{M}_k^1 \\
\mathbf{c} = \frac{1}{2} (\mathbf{U}^\top \mathbf{W}^{-1} \mathbf{U} - \mathbf{x}_{\text{fr},v}^T \mathbf{D}_{\text{fr},v}^{-1} \mathbf{x}_{\text{fr},v} - \mathbf{z}) + \ln(C_0)
\]

(40)

Where is:

\[
\mathbf{U} = \mathbf{H}^\top \mathbf{D}_n^{-1} \mathbf{y}_k + \mathbf{D}_\xi^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 \\
\mathbf{W} = (\mathbf{H}^\top \mathbf{D}_n^{-1} \mathbf{H} - \mathbf{D}_\xi^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{F}^\top \mathbf{D}_\xi^{-1} + \mathbf{D}_\xi^{-1})^{-1}
\]

The components \(\mathbf{D}_k, \mathbf{M}_k^1, \mathbf{M}_k^1\) are found iteratively as:

For \(i = 2;\)

For \(i = [3, k - v];\)

\[
\begin{align*}
\mathbf{D}_{v+2} &= (\mathbf{H}^\top \mathbf{D}_n^{-1} \mathbf{H} + \mathbf{D}_\xi^{-1} + \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{F})^{-1} \\
\mathbf{M}_{v+1} &= \mathbf{D}_\xi^{-1} \mathbf{F} \mathbf{D}_k^{-1} \mathbf{M}_k^1 \\
\mathbf{M}_{v+2} &= \mathbf{H}^\top \mathbf{D}_n^{-1} \mathbf{y}_{v+i-1} \\
\mathbf{R}_{v+2} &= \mathbf{F}^\top \mathbf{D}_k^{-1} \mathbf{F} \\
\mathbf{R}_{v+1} &= 0
\end{align*}
\]

(42)

Now let’s move on to finding the interpolation estimate \(\mathbf{x}_{\text{int},v}\) and the variance of the interpolation estimate \(\mathbf{D}_{\text{int},v}\) .

\[
\mathbf{P} = \mathbf{D}_{\text{int},v}^{-1}, \text{ therefore: } \mathbf{D}_{\text{int},v} = \mathbf{P}^{-1} : \\
\mathbf{q}^T = \mathbf{x}_{\text{int},v}^\top \mathbf{D}_{\text{int},v}^{-1}, \text{ therefore: } \mathbf{x}_{\text{int},v} = \mathbf{P}^{-1} \cdot \mathbf{q};
\]

Denote:

\[
\begin{align*}
\mathbf{D}_n^{\text{inv}} &= \mathbf{D}_n^{-1} \\
\mathbf{D}_\xi^{\text{inv}} &= \mathbf{D}_\xi^{-1}
\end{align*}
\]

(43)

Then the final algorithm for finding the estimate \(\mathbf{x}_{\text{int},v}\):

\[
\begin{align*}
\mathbf{x}_{\text{int},v} &= \mathbf{P}^{-1} \cdot \mathbf{q} \\
\mathbf{P} &= \mathbf{R}_k^1 + \mathbf{D}_{\text{fr},v}^{-1} - \mathbf{M}_k^T \mathbf{D}_k^T \mathbf{F} \mathbf{D}_\xi^{\text{inv}} \mathbf{W} \mathbf{D}_\xi^{\text{inv}} \mathbf{F} \mathbf{D}_k^T \mathbf{M}_k^1 - \mathbf{M}_k^{1T} \mathbf{D}_k \mathbf{M}_k^1 \\
\mathbf{q} &= \mathbf{D}_{\text{fr},v}^{-1} \mathbf{x}_{\text{fr},v} + \mathbf{M}_k^T \mathbf{D}_k^T \mathbf{F} \mathbf{D}_\xi^{\text{inv}} \mathbf{W} \mathbf{U} + \mathbf{R}_k^2 + \mathbf{M}_k^{1T} \mathbf{D}_k \mathbf{M}_k^2
\end{align*}
\]

(44)

Where is:
\[ U = H^T D_n^{inv} y_k + D_n^{inv} F D_k M_k^2 \]
\[ W = (H^T D_n^{inv} H + D_n^{inv} F D_k F^T D_n^{inv} + D_k^{inv})^{-1} \]

Estimates obtained from the Kalman filter:

**Extrapolation step:**
\[ \overline{D}_{flt} = F^T D_n^{-1} F + D_k \]
\[ \overline{x}_{flt} = F \hat{x}_{flt} \]

**Correction step:**
\[ D_{flt} = \overline{D}_{flt}^{-1} + H^T D_n^{inv} H \]
\[ \hat{x}_{flt} = \overline{x}_{flt} + D_{flt} H D_n^{inv} y \]

The components \( D_k, M_k^{i}, M_k^{2}, R_k^{i} \) and \( R_k^{2} \) are found iteratively as:

\[
\begin{align*}
D_{v+2} &= (H^T D_n^{inv} H + D_n^{inv} F^T D_n^{inv} F)^{-1} & D_{v+i} &= (D_{v+2} - D_k^{inv} F D_{v+i-1} F^T D_n^{inv})^{-1} \\
M_{v+2} &= D_n^{inv} F & M_{v+i} &= D_k^{inv} F D_{v+i-1} M_{v+i-1}^{1} \\
M_{v+i} &= H^T D_n^{inv} y_{v+i-1} + D_n^{inv} F D_{v+i-1} M_{v+i-1}^{2} \\
R_{v+2} &= F^T D_n^{inv} F & R_{v+i} &= R_{v+2} - M_{v+i-1}^T D_{v+i-1} M_{v+i-1}^{1} \\
R_{v+i} &= R_{v+i-1} + M_{v+i-1}^T D_{v+i-1} M_{v+i-1}^{2} \tag{47}
\end{align*}
\]

At each cycle of the algorithm, it is necessary to perform \( \delta + 3 \) matrix inversions. Let us set ourselves the goal of reducing the computational load of the algorithm represented by the equations (44), (45). The maximum number of matrix accesses occurs during iterative computation \( D_{v+i} \) (for \( i = [2, k-v] \)). Because \( D_{v+i} \) does not depend on incoming observations, then you can calculate the value of the matrices \( D_{v+i} \) at each step \( i = [2, k-v] \) in advance, save and use further. Thus, the computational load can be significantly reduced. It is also possible to pre-compute \( W \), which will save one more matrix inversion. It is logical to calculate in advance and \( R_k^{1} \) (since it is required to know only the matrix \( R_k^{1} \), and not the values of this matrix at each step \( i = [2, k-v] \)). However, in this case, additional memory resources must be used to store the following items:

- Matrix \( R_k^{1} \) with dimensions \( N*N \);
- Matrix \( W \) with dimensions \( N*N \);
- Matrix \( D_{v+i} \) with dimensions \( N*N \) at each step \( i = [2, k-v] \) (The number of matrices \( D_{v+i} \) is \( \delta \)).

In this case, the final algorithm for finding the estimate \( \hat{x}_{nv} \) is divided into two stages:

**Stage 1.** Calculating and storing matrix values \( R_k^{1}, W \) and \( D_{v+i} \) (is performed at the beginning of the algorithm or when changing \( D_n \) or \( D_{flt} \)): 

\[
W = (H^T D_n^{inv} H - D_n^{inv} F D_k F^T D_n^{inv} + D_k^{inv})^{-1} \tag{46}
\]

The components \( D_k \) and \( R_k^{1} \) are found iteratively as:

\[
\begin{align*}
\text{for } i &= 2; \\
\text{for } i &= [3, k-v]; \tag{47}
\end{align*}
\]
Stage 2. Finding the estimate \( \hat{x}_{iv,v} \) using the results of Stage 1.

\[
\hat{x}_{iv,v} = P^{-1} \cdot q
\]

\[
q = D_{iv,v}^{-1} \hat{\Phi}_{iv,v} + M_{iv,v}^{1} D_{iv,v}^{T} D_{iv,v}^{T} F W U R_{iv,v}^{1} + M_{iv,v}^{1T} D_{iv,v} F \hat{\Phi}_{iv,v} - M_{iv,v}^{1T} D_{iv,v} M_{iv,v}^{2}
\]

Where is:

\[
U = H^{T} D_{iv,v}^{T} M_{iv,v}^{2}
\]

The components \( M_{iv,v}^{1} \), \( M_{iv,v}^{2} \) and \( R_{iv,v}^{2} \) are found iteratively as:

<table>
<thead>
<tr>
<th>for ( i = 2 )</th>
<th>for ( i = [3,k-v] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_{iv,v}^{1} = D_{iv,v}^{T} F )</td>
<td>( M_{iv,v}^{2} = D_{iv,v}^{T} F D_{iv,v}^{-1} M_{iv,v}^{1} )</td>
</tr>
<tr>
<td>( M_{iv,v}^{2} = H^{T} D_{iv,v}^{-1} y_{v+i-1} )</td>
<td>( M_{iv,v}^{2} = H^{T} D_{iv,v}^{-1} y_{v+i-1} + D_{iv,v}^{T} M_{iv,v}^{2} )</td>
</tr>
<tr>
<td>( R_{iv,v}^{2} = 0 )</td>
<td>( R_{iv,v}^{2} = R_{v+i-1}^{2} + M_{v+i-1}^{T} D_{v+i-1} M_{v+i-1}^{2} )</td>
</tr>
</tbody>
</table>

As a result, the algorithm represented by equations (46)-(49) should make only 2 matrix accesses on each cycle of work.

**Checking the optimal linear interpolation algorithm.** The verification was carried out on a simulation model. The model is a system for estimating the three-dimensional Markov process of the phase of the radio signal:

\[
\varphi_{k} = \varphi_{k-1} + \omega_{k-1} T \\
\omega_{k} = \omega_{k-1} + a_{k-1} T \\
a_{k} = a_{k-1} \cdot e^{-\alpha T} + \xi_{k-1}
\]

Where is \( T = 10 \text{ ms} \) – sampling step; \( \alpha = 0.01 \text{ (1/sec)} \) – component fluctuation spectrum width \( \alpha_{k}, \xi_{k-1} \) - DWGN with zero mean and variance \( \sigma_{\alpha}^{2}(1-e^{-2\alpha T}) \), \( \sigma_{\alpha} = 15 \).

Reducing (50) to the form (3) as a result we obtain:

\[
\begin{pmatrix}
\varphi \\
\omega \\
\alpha
\end{pmatrix} =
\begin{pmatrix}
1 & T & 0 \\
0 & 1 & T \\
0 & 0 & e^{-\alpha T}
\end{pmatrix}
\begin{pmatrix}
n \\
0 \\
0
\end{pmatrix}
+
\begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
\begin{pmatrix}
\zeta \\
\xi
\end{pmatrix}

\[
D_{t} =
\begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & \sigma_{\alpha}^{2}(1-e^{-2\alpha T})
\end{pmatrix}
\]

The observation model is given as:

\[
y_{k} = \varphi_{k} + n_{k}
\]

Where is \( n_{k} \) - DWGN with zero mean and variance \( D_{n} \). To reduce (52) to the form (1)-(2) it is necessary to set \( H = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \).

Based on the observations \( y_{k} \) we will form an interpolation estimate of the process \( x_{v} \) with a delay value \( \delta = k - v \).

A comparison is made of the work of the presented linear interpolation algorithm and algorithms:

- filtration estimate obtained by the Kalman filter algorithm;
interpolation estimate according to the algorithm of two-way interpolation, set out in [2-5] (the algorithm works in fixed-delay interpolation mode \( \delta = k - \nu \));

interpolation estimate according to the interpolation algorithm set out in [12].

Evaluation criterion - sample standard deviation of phase estimation error \( \varphi_k \) (sample from 50000).

Algorithms were compared with the same implementations of the phase process and observation noise.

The results at \( D_n = 2.6e-6 \) and \( \delta = 10 \) are shown in the figure 1. Table 1 presents the values of the sample standard deviation of the phase estimation error for different algorithms. Thus, the use of interpolation instead of filtration algorithm made it possible to halve the sample standard deviation with a delay of the result by 10 cycles (100 ms).

**Table 1.** Values of the sample standard deviation of the phase estimation error for different algorithms at \( D_n = 2.6e-6 \) and \( \delta = 10 \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sample standard deviation [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman filter</td>
<td>( 0.819 \times 10^{-3} )</td>
</tr>
<tr>
<td>Optimal interpolation according to the algorithm described in [12]</td>
<td>( 0.414 \times 10^{-3} )</td>
</tr>
<tr>
<td>Optimal two-way interpolation according to the algorithm described in [2-5]</td>
<td>( 0.401 \times 10^{-3} )</td>
</tr>
<tr>
<td>Proposed optimal interpolation algorithm</td>
<td>( 0.400 \times 10^{-3} )</td>
</tr>
</tbody>
</table>

The results at \( D_n = 2.6e-6 \) and \( \delta = 50 \) are shown in the figure 2. Table 2 presents the values of the sample standard deviation of the phase estimation error for different algorithms. The use of the two-way interpolation algorithm and the linear interpolation algorithm presented in this paper made it possible to reduce the sample standard deviation by 2.3 times with a delay of the result by 10 cycles (500 ms). The optimal interpolation according to the algorithm presented in [12] is unstable when \( \delta = 50 \).

**Table 2.** Values of the sample standard deviation of the phase estimation error for different algorithms at \( D_n = 2.6e-6 \) and \( \delta = 50 \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sample standard deviation [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman filter</td>
<td>( 0.816 \times 10^{-3} )</td>
</tr>
<tr>
<td>Optimal interpolation according to the algorithm described in [12]</td>
<td>( 10.457 \times 10^{-3} )</td>
</tr>
<tr>
<td>Optimal two-way interpolation according to the algorithm described in [2-5]</td>
<td>( 0.362 \times 10^{-3} )</td>
</tr>
<tr>
<td>Proposed optimal interpolation algorithm</td>
<td>( 0.357 \times 10^{-3} )</td>
</tr>
</tbody>
</table>
Fig. 1. Phase estimation error for different algorithms at $D_n = 2.6e-6$ and $\delta = 10$
Fig. 2. Phase estimation error for different algorithms at $D_n = 2.6e-6$ and $\delta = 50$

6 Conclusions

As follows from the graphs in Fig.1-2, the proposed interpolation algorithm gives almost the same estimate as the algorithm presented in [2-5], which confirms the correctness of the conclusion made. The use of the proposed interpolation algorithm instead of filtration algorithm made it possible to reduce the sample standard deviation by a factor of 2 with a delay of the result by 10 cycles in the considered example of filtering the phase of the radio signal. Its advantage is the reduction of computational complexity in comparison with the algorithm [2-5]. The algorithm [2-5] requires $\delta + 3$ matrix accesses per cycle, while the proposed algorithm requires 2 matrix accesses per cycle. However, the fee is the use of additional memory resources to save pre-calculated elements.

References