INFORMATICA, 1990, Vol.1, No.2, 053–074

THE WEIGHT FUNCTION OF A SPACE–TIME AUTOREGRESSIVE FIELD IN SPACE R^2

Aloyzas KAPUSTINSKAS

Institute for Physical and Engineering Problems of Energy Research Lithuanian Academy of Sciences, 233684 Kaunas, Metalo St.4, Lithuania

Abstract. The weight coefficients calculation algorithm for an autoregressive random field, existing in a two–dimensional space and time, is proposed.

Key words: random field, autoregressive field, weight function.

1. Introduction. The properties of a space–time autoregressive (AR) random field, existing in a one-dimensional space R^1, were considered in the papers (Kapustinskas 1985a, 1985b, 1986a, 1986b, 1987, 1988, 1989a, 1989b). The properties of the weight function (WF) of such a field were considered in these papers, weight coefficients calculation methods were proposed and it was shown that they were useful for stability analysis and calculation of theoretical autocovariations of the field. However, there exist more complicated space-time AR fields. They function not in a one-dimensional but in a two-dimensional space R^2, i.e. on a plane.

The aim of this paper is to determine the WF structure of the AR field, existing in space R^2 and time, and to develop a weight coefficients calculation algorithm for such a field.

2. Statement of the problem. The space-time random AR field in space R^2 is described by such a difference equation

$$\xi_t^{xy} = \sum_{k=1}^{n_t} \sum_{i_x=-n'_x}^{n''_x} \sum_{i_y=-n'_y}^{n''_y} a_k^{i_x,i_y} \xi_{t-k}^{x+i_x,y+i_y} + g_t^{xy}, \qquad (1)$$

where t are discrete time moments ($t \in (-\infty, \infty)$), x, y are discrete values of the space coordinates ($x, y \in (-\infty, \infty)$), ξ_t^{xy} is the value of the field at point (x, y) and moment t, n_t is the order of the field with regard to coordinate t, $\{n'_x,\ n''_x\}$, $\{n'_y,\ n''_y\}$ is the order of the field with regard to the space coordinates x, y, respectively, $a_k^{i_x,i_y}$ are the parameters of the field, $\{g_t^{xy}\}$ is the sequence of independent normal random values with zero avarage and finite dispersion σ_g^2.

Let the order $\{n_t,\ n'_x,\ n''_x,\ n'_y,\ n''_y\}$ and parameters $a_k^{i_x,i_y}$ of field (1) be known. We shall determine the WF structure of the field, i.e. we shall consider at which points (x, y, t) the weight coefficients h_t^{xy} differ from zero. We shall also develop an algorithm for the calculation of nonzero h_t^{xy} at a certain time interval ($t = 1,\ 2, \ldots, T$).

3. Equation of WF. By WF of any dynamic system a transient process of the system, caused by a unit pulse input signal under zero initial conditions is called (Voronov, 1965). The random field, described by difference (1), is also a dynamic system whose input signal is a white noise field g_t^{xy}. Therefore random field (1) has a WF comprehended as a reaction of model (1) to the unit pulse input signal δ_t^{xy} at the point x, $y = 0$ and moment $t = 1$, i.e.

$$\delta_t^{xy} = \begin{cases} \delta_1^{00} = 1 & (x,\ y = 0,\ t = 1), \\ 0 & (x,\ y \neq 0,\ t \neq 1). \end{cases} \qquad (2)$$

It is supposed that there exist such zero initial conditions

$$\xi_t^{xy} = 0 \quad (x,\ y = 0, \pm 1, \ldots, \quad t = 0, -1, \ldots, -n_t + 1). \qquad (3)$$

Therefore it is easy to get the equation of weight coefficients from (1) replacing ξ_t^{xy} by h_t^{xy} and g_t^{xy} by δ_t^{xy}, i.e.

$$h_t^{xy} = \sum_{k=1}^{n_t} \sum_{i_x=-n'_x}^{n''_x} \sum_{i_y=-n'_y}^{n''_y} a_k^{i_x,i_y} h_{t-k}^{x+i_x,y+i_y} + \delta_t^{xy}. \tag{4}$$

Then the initial conditions are as folows:

$$h_t^{xy} = 0 \quad (x,\ y = 0, \pm 1, \ldots,\ \ t = 0, -1, \ldots, -n_t + 1). \tag{5}$$

This recurrent equation (4) together with initial conditions (5) is basic for determination of weight coefficients. The WF structure may be determined from this equation, too.

4. Structure of WF. It is easy to see from (3), (4), that $h_1^{xy} = \delta_1^{xy}$ as $t = 1$. Since $\delta_1^{xy} \neq 0$ only at the point $x,\ y = 0$, then

$$h_1^{xy} = \begin{cases} h_1^{00} = \delta_1^{00} = 1 & (x,\ y = 0), \\ 0 & (x,\ y \neq 0), \end{cases} \tag{6}$$

i.e. at the moment $t = 1$ there exists a single nonzero weight coefficient at the point $x,\ y = 0$ (point D_1 in Fig. 1). The others are equal to zero.

At the moment $t = 2$

$$h_2^{xy} = \sum_{k=1}^{n_t} \sum_{i_x=-n'_x}^{n''_x} \sum_{i_y=-n'_y}^{n''_y} a_k^{i_x,i_y} h_{t-k}^{x+i_x,y+i_y}, \tag{7}$$

because $\delta_2^{xy} = 0$.

All members $h_{2-k}^{x+i_x,y+i_y}$ $(k = 1, 2, \ldots, n_t)$ in (7) are equal to zero except for $h_1^{(\cdot)}$, which differs form zero, as $x + i_x = 0$ and $y + i_y = 0$. Indeed, according to the initial conditions (5), the weight coefficients $h_{2-k}^{(\cdot)} = 0$ for any $x,\ y$, as

$2-k \neq 1$, $-n'_x \leqslant i_x \leqslant n''_x$ and $-n'_y \leqslant i_y \leqslant n''_y$. The weight coefficients $h^{(\cdot)}_{2-k} = h^{00}_1 = 1$, as $2-k=1$ (i.e. as $k=1$), $x+i_x=0$ and $y+i_y=0$. Therefore

$$h^{xy}_2 = \begin{cases} h^{xy}_2 \neq 0 & (-n''_x \leqslant x \leqslant n'_x,\ -n''_y \leqslant y \leqslant n'_y), \\ 0 & \text{(in other cases)}, \end{cases} \tag{8}$$

i.e. the nonzero values h^{xy}_2 are inside the rectangle D_2 (Fig. 1).

In a similar way it can be shown that at moments $t = 3,\ 4$

$$h^{xy}_3 = \begin{cases} h^{xy}_3 \neq 0 & (-2n''_x \leqslant x \leqslant 2n'_x,\ -2n''_y \leqslant y \leqslant 2n'_y), \\ 0 & \text{(in other cases)}, \end{cases} \tag{9}$$

$$h^{xy}_4 = \begin{cases} h^{xy}_4 \neq 0 & (-3n''_x \leqslant x \leqslant 3n'_x,\ -3n''_y \leqslant y \leqslant 3n'_y), \\ 0 & \text{(in other cases)}, \end{cases} \tag{10}$$

i.e. the nonzero values h^{xy}_3, h^{xy}_4 at moments $t = 3,\ 4$ are inside the rectangles D_3, D_4, respectively (Fig. 1).

At whichever moment t

$$h^{xy}_t = \begin{cases} h^{xy}_t \neq 0 & (-n''_x(t-1) \leqslant x \leqslant n'_x(t-1), \\ & -n''_y(t-1) \leqslant y \leqslant n'_y(t-1)), \\ 0 & \text{(in other cases)}, \end{cases} \tag{11}$$

i.e. the nonzero values h^{xy}_t at moment t are inside the rectangle D_t (Fig. 1).

Hence it follows that the structure of WF of the field (1) is defined by (11), i.e. the nonzero weight coefficients are inside a polyhedral angle in a space $(x,\ y,\ t)$. The edges of the angle are four straight lines, starting from the point $(x,\ y,\ t) = (0,\ 0,\ 1)$ and crossing the tops of rectangles $D_2,\ D_3, \dots, D_t$ (Fig. 1). The coefficients h^{xy}_t outside this angle are equal to zero.

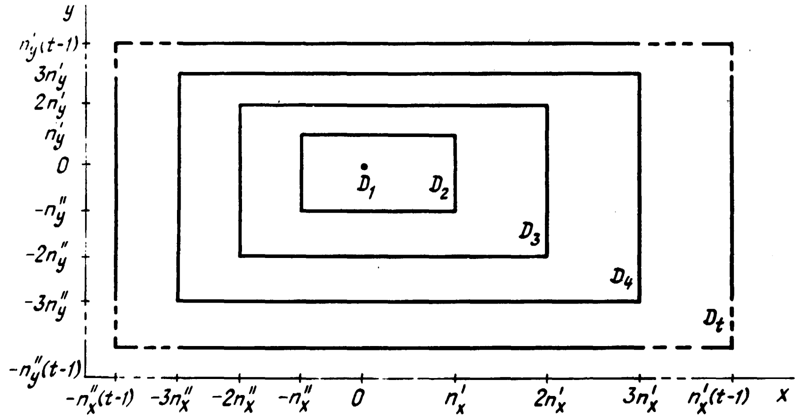

Fig. 1. Rectangle areas $D_1, D_2, \dots, D_t$, inside of which the nonzero weight coefficients h_t^{xy} at the moments $1, 2, \dots, t$ exist

5. Weight coefficient calculation algorithm. Above the structure of WF of the field (1) was determined, i.e. the area in the space $(x,\ y,\ t)$, inside of which coefficients h_t^{xy} have nonzero values. Now we shall consider the weight coefficients calculation problem, i.e. we shall develop the algorithm for the calculation of nonzero h_t^{xy} at moments $t = 1,\ 2, \dots, T$. The basis of the algorithm is a recurrent (4) with initial conditions (5). The nonzero coefficients at moments $t = 1,\ 2, \dots, T$, in accordance with the WF structure, are inside rectangles $D_1,\ D_2, \dots, D_T$ (Fig. 1), the area of which insreases with time. The greatest area has rectangle D_T, where $-n''_x(T-1) \leqslant x \leqslant n'_x(T-1)$, $-n''_y(T-1) \leqslant y \leqslant n'_y(T-1)$. According to (4) the coefficient h_t^{xy} at the point $(x,\ y)$ and moment t is composed of some number of weight coefficients, surrounding the considered point $(x,\ y)$ at past time moments. Therefore for the calculation of h_t^{xy} it is necessary to form a certain three-dimensional (in accordance with coordinates $x,\ y,\ i_t$) array $D_{x,\,y,\,i_t}$ in the main memory of a computer for every moment $t = 1, 2, \ldots, T$ and to store in it values h_t^{xy}, necessary for weight coefficients calculation at a current time moment for every point (x, y) of intervals $-n_x''(t-1) \leqslant x \leqslant n_x'(t-1)$, $-n_y''(t-1) \leqslant y \leqslant n_y'(t-1)$. This array can be represented as a parallelepiped in space (x, y, i_t) (Fig. 2a), at every discrete point of which the value of the corresponding coefficient h_t^{xy} is disposed. We shall determine the volume of this parallelepiped. Since being the greatest among the rectangles $D_1, D_2, \ldots, D_T$ is rectangle D_T (Fig. 1), it should be put as the base of parallelepiped D. However, at first the edges of D_t must be extended by a value of model order alongside axis x and y. This extension is necessary for the following reasons. While calculating by (4) the coefficients h_t^{xy} at the points lying on the left-side edge of rectangles D_t $(t = T)$, it is necessary to know the weight coefficients at moments $T-1, T-2, \ldots, T-n_t$ at points $(x-1, y), (x-2, y), \ldots, (x-n_x', y)$. Hence, it is necessary to extend rectangle D_T by a value n_x' in the direction of the negative x-axis. In the same way it can be shown that the rectangle D_T must be extended by values n_x'', n_y', n_y'' in the direction of the positive x-asis, negative and positive y-asis, respectively. Therefore the base of parallelepiped D occupies such an area: $-n_x''(T-1) - n_x' \leqslant x \leqslant n_x'(T-1) + n_x''$, $-n_y''(T-1) - n_y' \leqslant \leqslant x \leqslant n_y'(T-1) + n_y''$. The hight of the parallelepiped occupies an interval $1 \leqslant i_t \leqslant n_t + 1$. It is convenient to represent the parallelepiped D in the form of a number of layers $i_t = 1, 2, \ldots, n_t + 1$ (Fig. 2b). The values of weight coefficients at a current moment t are stored into the first layer, those at a moment $t-1$ – into the second layer, ect.

Calculation of the weight coefficients h_t^{xy} is realized by such an algorithm.

Step 0. The array D is cleaned, i.e. to all layers of parallelepiped D zeroes are stored.

Step 1. Variable t is assigned a unit value. The value

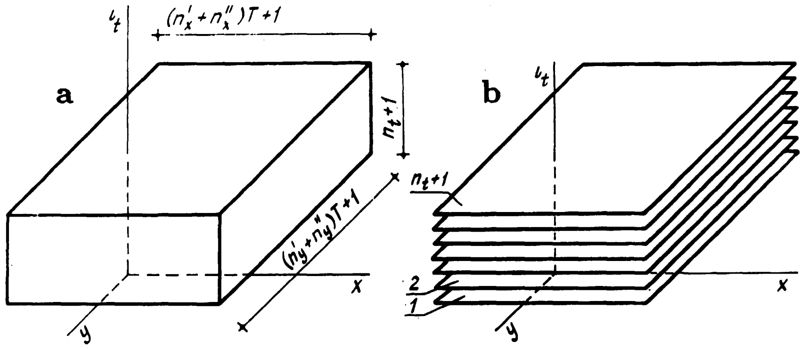

Fig. 2. Representation of array D as a parallelepiped not divided (a) or divided (b) into layers

$h_1^{00} = 1$ is stored to the point $x,\ y = 0$ of the first layer and is led out to the printer of a computer.

Step 2. Variable t is incremented by one.

Step 3. The n_t–layer is transferred to the $(n_t + 1)$ –layer, the $(n_t - 1)$ –layer – to the n_t –layer, ect. The first layer is cleaned.

Step 4. The values h_t^{xy} at a current time moment in the area $-n_x''(t-1) \leqslant x \leqslant n_x'(t-1),\ -n_y''(t-1) \leqslant y \leqslant n_y'(t-1)$ are calculated by (4) and transferred to the corresponding points of the first layer of array D.

Step 5. The weight coefficients from the above area of the first layer are led out to the printer of a computer.

Step 6. If $t < T$, then we return to step 2. In other cases calculations are ended.

Thus the above algorithm enables us to calculate the values of nonzero weight coefficients h_t^{xy}. However, it can be used only for relatively small values of T. Indeded, the volume of array D is determined as follows:

$$V' = [(n_x' + n_x'')T + 1][(n_y' + n_y'')T + 1](n_t + 1). \qquad (12)$$

$V' = 10^7$ as n'_x, n''_x, n'_y, $n''_y = 5$, $n_t = 9$ and $T = 100$, i.e. this algorithm requires a significant amount of the main memory.

Therefore it is worthwhile developing such modifications of the algorithm, which would require a smaller amount of the main memory for the same values of the field order and parameter T.

It is possible to solve this problem when storing array D (all or part of it) in the external computer memory and forming an array W (smaller than D) in the main memory. This can be realized by variuos methods. We shall consider some of them.

6. The first modification of the algorithm. The equation (4) can be rewritten in the following way

$$h_{tk}^{xy} = \sum_{i_x=-n'_x}^{n''_x} \sum_{i_y=-n'_y}^{n''_y} a_k^{i_x,i_y} h_{t-k}^{x+i_x,y+i_y} + \delta_t^{xy}, \qquad (13)$$

$$(k = 1,\ 2, \ldots, n_t),$$

$$h_t^{xy} = \sum_{k=1}^{n_t} h_{tk}^{xy}. \qquad (14)$$

When k is fixed, at first the intermediate values h_{tk}^{xy} in the area $-n''_x(t-1) \leqslant x \leqslant n'_x(t-1)$, $-n''_y(t-1) \leqslant y \leqslant n'_y(t-1)$ are calculated and then these values are summed according to (14). As a result, we get the weight coefficients h_t^{xy} at a current moment t. Under this approach it is necessary to form a three-dimensional array $W_1(x,\ y,\ i_t)$ in the main memory. The volume of W_1 is equal to that of the first two layers ($i_t = 1,\ 2$) of array D (Fig. 3). The volume W_1 is

$$V' = [(n'_x + n''_x)T + 1][(n'_y + n''_y)T + 1]. \qquad (15)$$

In the external memory an array $D_1(x,\ y,\ i_t)$, consisting of the second, third, etc. ($i_t = 2,\ 3, \ldots, n_t + 1$) layers of array D (Fig. 3), is stored. The volume of D_1 is

$$V'' = [(n'_x + n''_x)T + 1][(n'_y + n''_y)T + 1]n_t. \qquad (16)$$

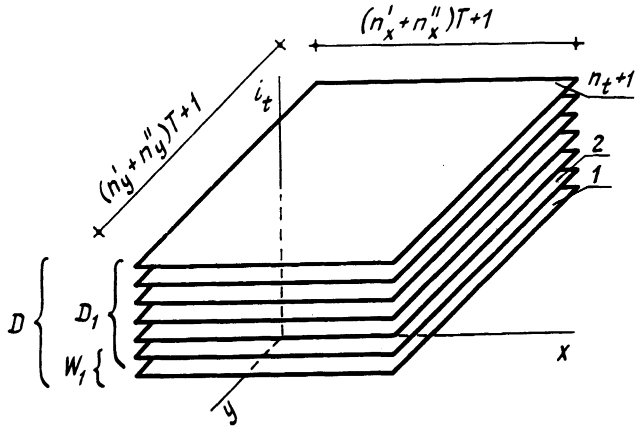

Fig. 3. Division of array D in to array D_1 and W_1 in the first modification of WF calculation algorithm

Calculations of the nonzero weight coefficients h_t^{xy} ($t = 1,\ 2, \ldots, T$) are realised by such a scheme.

Step 0. Arrays W_1 and D_1 are cleaned.

Step 1. Variable t is assigned a unit value. The value $h_1^{00} = 1$ is stored to the point $x,\ y = 0$ of the first layer of W_1 and is led out to the printer.

Step 2. The first layer of W_1 is transferred to the second layer of array D_1, stored in the external memory.

Step 3. Variable t is incremented by one. A variable i_t is assigned a unit value.

Step 4. Variable i_t is incremented by one. The first layer of W_1 is cleaned.

Step 5. The i_t–layer of D_1 is transferred from the external memory to the second layer of W_1.

Step 6. By (13) the intermediate values h_{tk}^{xy} in the area $-n_x''(t-1) \leqslant x \leqslant n_x'(t-1)$, $-n_y''(t-1) \leqslant y \leqslant n_y'(t-1)$ are calculated. They are summed with the values from the first layer of W_1 according to (14). The results are transferred to the corresponding points of this layer.

Step 7. If $i_t < n_t + 1$, we return to step 4, in other cases – to step 8.

Step 8. The values of the weight coefficients from the area $-n_x''(t-1) \leqslant x \leqslant n_x'(t-1)$, $-n_y''(t-1) \leqslant y \leqslant n_y'(t-1)$ of the first layer of W_1 are led out to the printer.

Step 9. The n_t–layer of D_1 is transferred to the (n_t+1)-layer, the (n_t-1)–layer – to the n_t–layer. etc.

Step 10. The first layer of W_1 is stored in the external memory, i.e. in the second layer of D_1.

Step 11. If $t < T$, we return to step 5. In other cases calculations are ended.

In the basic algorithm it is necessary to store the whole array D, whose size is determined by (12), in the main memory. For the first modification of the algorithm it is necessary to store in the main memory array W_1 of the volume, determined by (15). It is evident from (15) and (12) that this modification enables to decrease the required amount of the main memory $(n_t+1)/2$ times.

7. The second modification of the algorithm. A still less amount of the main memory is needed, when using an algorithm, based on the replacement of (4) by the two following equations:

$$h_{t,i_x}^{xy} = \sum_{k=1}^{n_t} \sum_{i_y=-n_y'}^{n_y''} a_k^{i_x,i_y} h_{t-k}^{x+i_x,y+i_y} + \delta_t^{xy} \qquad (17)$$

$$(i_x = \overline{-n_x',\ n_x''}),$$

$$h_t^{xy} = \sum_{i_x=-n'_x}^{n''_x} h_{t,i_x}^{xy}. \tag{18}$$

This means that for every current moment t ($t = 1, 2, \ldots, T$) and every x, y from the area $-n''_x(t-1) \leqslant x \leqslant \leqslant n'_x(t-1)$, $-n''_y(t-1) \leqslant y \leqslant n'_y(t-1)$ the intermediate values h_{t,i_x}^{xy} are calculated for a fixed i_x by (17). Then these values are summed by (18) and, as a result, we get the weight coefficients h_t^{xy}. For the realization of this, it is necessary to divide array D not into layers but into cuts, represented in Fig. 4a. The array D, divided into cuts, will be denoted by D_2. In the array D_2 there are $(n'_x + n''_x)T + 1$ cuts with numbers $l_x = l'_x,\ l'_x + 1, \ldots, l''_x$, where $l'_x = -n''_x(T-1) - n'_x,\ l''_x = = n'_x(T-1) + n''_x$. These cuts are stored in the external memory of a computer. In the main memory a two – dimensional array $W_2(y,\ i_t)$ is formed, divided into zero, first, …, (n_t+1)–st layers ($i_t = 0,\ 1, \ldots, n_t+1$), i.e. into n_t+2 layers (Fig. 4b). Array W_2 is equal to whichever cut of array D_2, extended by one 0- layer.

Therefore the volume of the array is

$$V' = [(n'_y + n''_y)T + 1](n_t + 2). \tag{19}$$

Calculations of the nonzero weight coefficients h_t^{xy} are realized in such a way.

Step 0. Arrays W_2 and D_2 are cleaned.

Step 1. Variable t is assigned a unit value. The value h_1^{00} is stored to the point $x, y = 0$ of the first layer of array W_1 and is led out to the printer.

Step 2. Array W_2 (without 0-layer) is transferred to the 0-cut ($l_x = 0$) of array D_2, in the external memory.

Step 3. Variable t is incremented by one. Variable x is assigned value $-n''_x(t-1) - 1$.

Step 4. Variable x is incremented by one. The 0-layer of W_2 is cleaned. Variable i_x is assigned a value $-n'_x - 1$.

Step 5. Variable i_x is incremented by one.

Step 6. The $(x + i_x)$–cut of array $\dot{D}_2$ is transferred from the external memory to array W_2, starting from the first layer.

Step 7. Intermediate values h^{xy}_{t,i_x} from the interval $-n''_y(t-1) \leqslant y \leqslant n'_y(t-1)$ are calculated by (17). They are summed with the values from the first layer of W_2 according to(18). The results are transferred to the corresponding points of this layer.

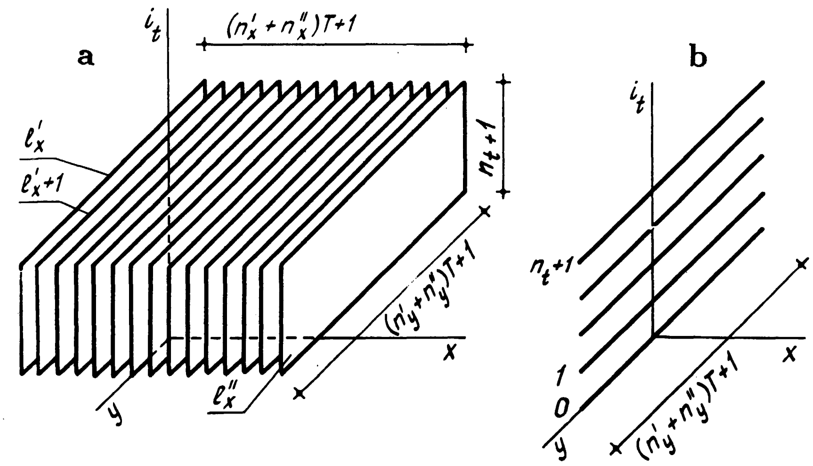

Fig. 4. a, b – arrays D_2 and W_2 in the second modification of the WF calculation algorithm, respectively

Step 8. If $i_x < n''_x$, we return to step 5, in other cases – to step 9.

Step 9. The values of the 0-layer of W_2 are led out to the printer, i.e. the values of weight coefficients for current t, x and for all y from the interval $-n''_y(t-1) \leqslant y \leqslant n'_y(t-1)$ are printed. The 0–layer of W_2 is transferred to the first layer. Array W_2 (without the 0–layer) is transferred to the x–cut of D_2 into the external memory.

Step 10. If $x < n'_x(t-1)$, we return to step 4, in other cases–to step 11.

Step 11. Variable x is assigned value $-n''_x(t-1)-1$.

Step 12. The x–cut of D_2 is transferred to array W_2, starting from the first layer. The n_t–layer of W_2 is transferred to the (n_t+1)–layer, the (n_t-1)–layer to the n_t–layer, etc. Array W_2 (without the 0–layer) is transferred to the x–cut of D_2 in the external memory.

Step 13. If $x < n'_x(t-1)$, we return to step 11, in other cases calculations are ended.

The volume of array W_2 $[(n'_x + n''_x)T + 1](n_t+1)/(n_t+$ $+2)$ times smaller than that of array D. Therefore the second modification of the algorithm enables to decrease the required amount of the main memory the same number of times.

8. The third modification of the algorithm. It is possible to make one more step towards decreasing the necessary resources of the main memory. The equation (4) can be replaced by the following three equations

$$h^{xy}_{t,i_x,k} = \sum_{i_y=-n'_y}^{n''_y} a^{i_x,i_y}_{t-k} h^{x+i_x,y+i_y}_{t-k} + \delta^{xy}_t \tag{20}$$

$$(i_x = \overline{-n'_x,\ n''_x},\ k = \overline{1,\ n_t}),$$

$$h^{xy}_{t,i_x} = \sum_{k=1}^{n_t} h^{xy}_{t,i_x,k} \tag{21}$$

$$(i_x = \overline{-n'_x,\ n''_x}),$$

$$h^{xy}_t = \sum_{i_x=-n'_x}^{n''_x} h^{xy}_{t,i_x}. \tag{22}$$

Then array D_3 must be stored in the external memory. Array D_3 is the same array D but it is divided into layers and cuts (Fig.5a). The number of cuts in array D_3 is the same as in D_2, i.e. $(n'_x + n''_x)T + 1$ $(l_x = l'_x, l'_x + 1, \ldots, l''_x$, where $l'_x = -n''_x(T-1) - n'_x$, $l''_x = n'_x(T-1) + n''_x)$. The number of layers in every cut of array D_3 is the same as in array D, i.e. $n_t + 1$ $(i_t = 1, 2, \ldots, n_t + 1)$. The total number of layers in array D_3 is equal to $[(n'_x + n''_x)T + 1](n_t + 1)$.

For this modification of the algorithm it is necessary to store the two–dimensional array $W_3(y, i_t)$ in the main memory. This array is equal to the two neighbouring layers of whichever cut of array D_3 (Fig.5b).

Calculations of the nonzero weight coefficients h_t^{xy} are realized by the following scheme.

Step 0. Arrays W_3, D_3 are cleaned.

Step 1. Variable t is assigned an unit value. The value $h_1^{00} = 1$ is stored to the point $x, y = 0$ of the first layer of W_3 and is led out to the printer.

Step 2. The first layer of W_3 is transferred to the second layer of the 0–layer of the 0–cut $(l_x = 0)$ of D_3.

Step 3. Variable t is incremented by one. Variable x is assigned value $-n''_x(t-1) - 1$.

Step 4. Variable x is incremented by one. The first layer of W_3 is cleaned. Variable i_x is assigned value $-n'_x - 1$.

Step 5. Variable i_x is incremented by one. Variable i_t is assigned an unit value.

Step 6. Variable i_t is incremented by one.

Step 7. The i_t–layer of the $(x + i_x)$–cut of D_3 is called from the external memory to the second layer.

Step 8. The intermediate values $h_{t,i_x,k}^{xy}$ at the interval $-n''_y(t-1) \leqslant y \leqslant n'_y(t-1)$ are calculated by (20). They are summed with the values from the first layer of W_3 in according to (21) and (22). The results are stored to the corresponding points of this layer.

Step 9. If $i_t < n_t + 1$, we return to step 6, in other cases – to step 10.

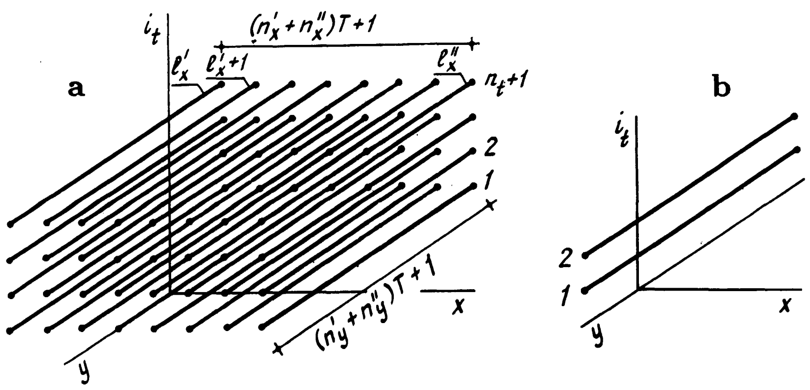

Fig. 5. a, b – arrays D_3 and W_3 of the third modification of the WF calculation algorithm, respectively

Step 10. If $i_x < n_x''$, we return to step 5, in other cases – to step 11.

Step 11. The values of the first layer of W_3 are led out to the printer, i.e. the values of the weight coefficients for current t, x and for every y from the internal $-n_y''(t-1) \leqslant y \leqslant n_y'(t-1)$ are printed. The first layer of W_3 is transferred to the first layer of the x–cut of D_3 in the external memory.

Step 12. If $x < n_x'(t-1)$, we return to step 4, in other cases–to step 13.

Step 13. Variable x is assigned value $-n_x''(t-1)-1$.

Step 14. Variable x is incremented by one. Variable i_t is assigned value $n_t + 1$.

Step 15. Variable i_t is incremented by one. The i_t–layer of the x–cut of D_3 is transferred to the second layer of W_3. The second layer of W_3 is transferred to the $(i_t + 1)$–layer of the x–cut of D_3.

Step 16. If $i_t > 1$, we return to step 15, in other cases – to step 17.

Step 17. If $x < n''_x(t-1)$, we return to step 14, in other cases – to step 18.

Step 18. If $t < T$, we return to step 3, in other cases calculations are ended.

The volume W_3 is

$$V' = 2[(n'_y + n''_y)T + 1]. \tag{23}$$

It is easily seen from (12) and (13) that the volume of W_3 is $[(n'_x + n''_x)T + 1](n_t + 1)/2$ times smaller than that of arraay D. Therefore the above modification of the algorithm requires the same number of times smaller volume of the main memory than the basic algorithm.

9. Analysis of the algorithms. It is necessary to refer to the external memory for arrays D_1, D_2 or D_3 when calculations are realized by the first–third modifications of the algorithm. The number of these references can be determined from the analysis of the shemes of the algorithms. It is easily seen that one is obliged to refer to the external memory in such cases: 1) for cleaning array D, i.e. D_1, D_2 or D_3, 2) for calling separate layers or cut of arrays D in the process of weight coefficients calculation, 3) for storing weight coefficients to array D, 4) for regrouping the layers of array D. Thus the total number of references is

$$K = K_v + K_s + K_h + K_p, \tag{24}$$

where K_v, K_s, K_h, K_p is the number of references in the above cases.

Deteremination of K_v, K_s, K_h, K_p will not be considered in detail because of a restricted volume of the paper. The total number of references for the first–third modifications of the algorithm is

$$K_1 = (3n_t + 1)T - 2n_t, \tag{25}$$

$$K_2 = c_2'T^2 + c_1'T + c_0', \tag{26}$$

$$K_3 = c_2''T^2 + c_1''T + c_0'', \tag{27}$$

where

$$c_0' = 1 - (n_x' + n_x'')^2, \tag{28}$$
$$c_1' = (n_x' + n_x'' + 1)(n_x' + n_x'')/2 + 1, \tag{29}$$
$$c_2' = (n_x' + n_x'')(n_x' + n_x'' + 3)/2, \tag{30}$$
$$c_0'' = 1 - n_t(n_x' + n_x'' + 2), \tag{31}$$
$$c_1'' = \{1 + 4n_t + (n_x' + n_x'' + 1)[1 - n_t(n_x' + n_x'' - 2)]\}/2, \tag{32}$$
$$c_2'' = \{(n_x' + n_x'')[1 + n_t(n_x' + n_x'' + 3)]\}/2. \tag{33}$$

$K_0 = 0$, since the basic algorithm does not use the external memory.

It is evident from (25)–(33) that the number of references to the external memory depends both on the duration of time interval T and on the order of the field. The number K increases by the linear law for the first modification and by the quadratic law for other modifications of the algorithm with an increase of T. Also K increases with an increase of the field order. For the first modification this dependence is straight–line, for the second–quadratic and for the third–cubic. In the case when

$$n_x' + n_x'' = n_y' + n_y'' = n_t + 1 = n, \tag{34}$$

these dependences are represented in Fig.6. It can be seen that the third modification of the algorithm riequires the greatest number of references to the external memory, and the first modification – the smallest number (the second modification takes the intermediate place), i.e.

$$K_3 > K_2 > K_1 > K_0 = 0. \tag{35}$$

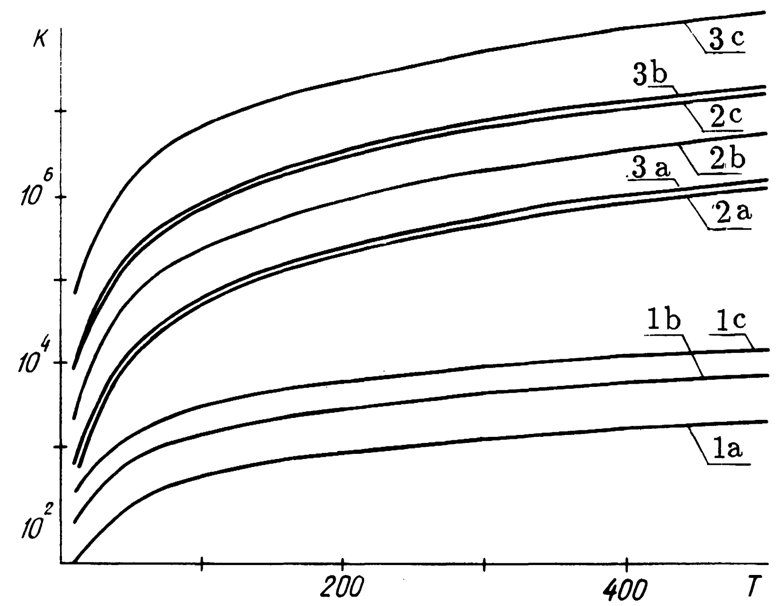

Fig. 6 Dependence of the number of references to the external memory $K = f(T)$ as $n = 2, 5, 10(a, b, c)$. 1–3 – for the first, second and third modifications of the algorithm, respectively

This means that the basic algorithm requires the smallest, and the third modification – the greatest amount of computer time, i.e. the basic algorithm is the most fast and the third modification – the slowest algorithm.

It is easy to see from (12), (16), (19), (23) that the volume of the required main memory increases with increase of interval T. It increases by qudratic dependence for the basic algorithm and the first modification, and by the linear dependence – for other modifications. Also V' increases with an increase of the field order. For the basic algorithm this dependence is cubic, for the first and second modifications – quadratic, and for the third–straight–line. In the case (34) these dependences are represented in Fig. 7. Above it was determined (it can be seen from Fig. 7, too) that the basic algorithm uses the greatest and the third modification–the smallest amount of the main memory, i.e.

$$V_0' > V_1' > V_2' > V_3', \tag{36}$$

where V_i' is the volume V' of an i – algorithm.

Any computer has a limited main memory. Let us use a computer with volume V of the main memory. Let T and the model order be such that $V_0' \leqslant V$. In this case we can use any algorithm for weight coefficients calculation. However, it is no use using the first–third modifications of the algorithm, because they are less fast than the basic algorithm. Let $V_0' >$ $> V$. In this case the basic algorithm cannot be used since its main memory is too small. In the case $V_0' > V$ and $V_1' \leqslant V$ we can use the first modification of the algorithm, etc. Therefore we can formulate such a rule for choosing an algorithm:

$$i = \begin{cases} 0 & (V_0' \leqslant V), \\ 1 & (V_0' > V, \ \ V_1' \leqslant V), \\ 2 & (V_0', V_1' > V, \ \ V_2' \leqslant V), \\ 3 & (V_0', V_1', V_2' > V, \ \ V_3' \leqslant V), \end{cases} \tag{37}$$

where i is the number of the algorithm ($i = 0$ – basic algorithm, $i = 1 \div 3$ – its first–third modifications)

If $V_0', V_1', V_2', V_3' > V$, then no algorithm can be used. In this case interval T should be decreased.

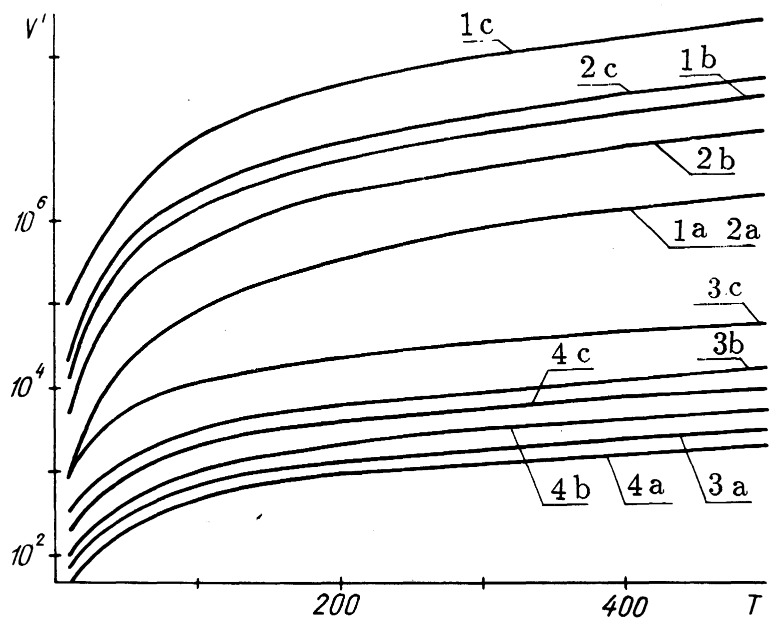

Fig. 7 Dependence of the necessary volume of the main memory of a computer $V' = f(T)$ as $n = 2, 5, 10$ (a, b, c). 1–4 – for the basic algorithm and the first, second and third modifications, respectively

The volume of the necessary external memory for first modification of the algorithm is

$$V'' = [(n'_x + n''_x)T + 1][(n'_y + n''_y)T + 1]n_t, \qquad (38)$$

and for the second and third modifications

$$V'' = [(n'_x + n''_x)T + 1][(n'_y + n''_y)T + 1](n_t + 1). \qquad (39)$$

10. Conclusions. The weight cofficients of a space – time AR field in space R^2, described by (1), are determined by a recurrent equation (4) and zero initial conditions (5). The WF structure of this field is described by (11), i.e. nonzero weight coefficients are inside a certain polyhedral angle in a space (x, y, t). The edges of this angle are four straight lines, starting from the point $(x, y, t) = (0, 0, 1)$. The developed basic algorithm and its three modifications can be used for weight coefficients calculation at a certain time interval T. The greatest volume of computer main memory is required by the basic algorithm, the smallest – by the third modification (the others are in the intermediate place). The most fast is the basic algorithm, the slowest is its third modification. The rule (37) can be used for choosing of the most fast algorithm for a certain interval T and the field order.

REFERENCES

Voronov, A.A. (1965). *Automatic control.* Part 1. Energia, Moscow. 396pp. (in Russian).

Kapustinskas, A.J. (1985 a). Identification of autoregressive random fields (1. The change of a first order autoregressive field to a moving average field). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **2**(147), 63–72 (in Russian).

Kapustinskas, A.J. (1985 b). The same, (2. Covariation equations for first order models). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **3**(148), 81–90 (in Russian).

Kapustinskas, A.J. (1986 a). The same, (3. Structure of autocovariations for a first order model. *Works of the Academy of Sciences of the Lithuanian SSR,* Ser.B **4**(155), 125–133 (in Russian).

Kapustinskas, A.J. (1986 b). The same, (4. Stability conditions for a first order model). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **5**(156), 83–94 (in Russian).

Kapustinskas, A.J. (1987). The same, (5. Parameter estimation of models). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **6**(163), 117–124 (in Russian).

Kapustinskas, A.J. (1988). The same, (6. Determination of the type of first order models). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **1**(164), 131–137 (in Russian).

Kapustinskas, A.J. (1989 a). The same, (7. Formulae of theoretical autocovariances of one–and two–parametric models). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **1**(170), (in Russian).

Kapustinskas, A.J. (1989 b). The same, (8.A Theoretical autocovariances calculation algorithm for n–order models). *Works of the Academy of Sciences of the Lithuanian SSR,* Ser. B **3**(172), (in Russian).

Received January 1990

A. Kapustinskas in 1972 maintained a candidate thesis at the Kaunas Polytechnic Institute. At present he is a senior researcher of Adaptive System Department at the Institute for Physical and Engineering Problems of Energy Research, Lithuanian Acad. of Sci. His research interests are identification of random processes and fields.