Crossbar Array Programming Using Piecewise-Constant Signals

G. A. Beskhlebnova[1], V. B. Kotov[2]

[1]Scientific Research Institute for System Analysis of the National Research Centre Kurchatov Institute, Moscow, Russia, beskhlebnova@niisi.ras.ru;

[2]Scientific Research Institute for System Analysis of the National Research Centre Kurchatov Institute, Moscow, Russia, v1111111k1111@gmail.com

Abstract. To program a crossbar array, we need to adjust the resistor conductance using a limited number of control signals, which are voltages applied to the crossbar lines. Since the number of lines is significantly smaller than the number of resistors, this is a multi-step procedure. At each step, the conductances of the selected resistors are adjusted. The number of such resistors is no greater than the number of control signals. This inevitably changes the conductivity of some half-selected resistors, too. These unwanted changes must be compensated for. We examined a crossbar programming procedure using high-frequency piecewise-constant control signals. Our analysis involved a simple resistive element model. We demonstrated that an arbitrary (within known limits) conductance matrix can be programmed. At each step, a row or column of the crossbar array is generated or adjusted. We discussed the feasibility and convenience of such a procedure.

Keywords: variable resistor, piecewise-constant signal, crossbar array, conductance matrix

1. Introduction

Large crossbar arrays can become the basic building blocks of neuromorphic systems. They perform vector–matrix multiplication, the most computationally intensive operation in neural computing. The values of the multiplier matrix elements are represented by the conductances of the crossbar array resistors [1]. To use he same crossbar array for multiplication by different matrices, the resistors must be variable and controllable. Memristors are variable resistors whose resistance changes based on current [2-4]. They are commonly used due to their simplicity, compact size, and low energy consumption.

Programming large crossbar arrays is a challenging problem [1, 5]. It is impossible to control each resistor individually since their number is huge. In crossbars, two conductors connect to not one but multiple resistors. In crossbars, each conductor connects to multiple resistors, not just one. Therefore, a step-by-step programming procedure is required. With each step, a row or column of the crossbar is programmed. Using constant (more precisely, unipolar) voltage signals for crossbar programming leads to significant difficulties: the conductances of both selected and many half-selected resistors are affected. The direction of these undesirable changes varies for each half-selected resistor, making it difficult to compensate for them. These difficulties are often circumvented by assuming a threshold behavior of the conductance [2, 3, 6]. However, this does not work in real life. Nature does not always meet the expectations of scientists.

To achieve easily compensable changes in half-selected resistors, we can apply a modulated high-frequency signal to program the crossbar [7]. The use of high-frequency harmonic signals for crossbar programming is discussed in [8, 9]. It is shown that for unidirectional variable resistors, applying alternating harmonic voltages as control signals allows for programming virtually any (within certain limits) crossbar.

Other periodic high-frequency signals can also be used. Such signals should be sufficiently simple to generate to be practical. Piecewise-constant signals stand out as they can have only a few values. We studied the application of such signals to crossbar programming. We assumed that applying a single period of the control voltage produces an insignificant change in the variable resistor's conductance; multiple periods are required to program the crossbar (i.e., change the resistor conductance). It means that a high-frequency signal is needed. The signal values must be both positive and negative. Otherwise, it is a unipolar signal similar to a constant signal in its effect. We also assumed that the variable resistor is represented by a simple resistor element model [10].

2. Crossbar Programming Equations

The conductivity G and resistance R=1/G of a simple resistor element are expressed by only one state variable x: $G=G(x)$. We assumed that the state variable ranges from 0 to 1. The high-resistance state ($x = 0$) corresponds to the maximum resistance, and the low-resistance state ($x = 1$) corresponds to the minimum resistance. The change in the state variable is described by equation [10]

$$\frac{dx}{dt} = F(x,u), \quad (1)$$

where u is the voltage applied to the resistor. Note that, in general, the right-hand side of equation (1) depends on both the resistor voltage u and current I. Using Ohm's law (it can be applied to resistors by definition)

$$u = R(x)I \quad (2)$$

we can express I in terms of u and x as equation (1).

Let us express the function F () as

$$F(x,u) = F_0(x) + F^+(x,u) + F^-(x,u) \quad (3)$$

Where the term $F_0(x) = F(x, u = 0)$ describes the spontaneous change of the resistor toward its high-resistance state in the absence of any control signal; the term $F^+(x,u) = \theta(u)(F(x,u) - F_0(x))$ represents the tendency of the state variable x to increase under a positive voltage; the term $F^-(x,u) = \theta(-u)(F(x,u) - F_0(x))$ represents the tendency of x to decrease (accelerated relaxation) under a negative voltage. (Here $\theta(x)$ is the Heaviside function).

If $F^+(x,u)$ ($F^-(x,u)$) vs. x weakly depends on u, we can use factorization:

$$F^+(x,u) = F_x^+(x)F_u^+(u), F^-(x,u) = \\ = F_x^-(x)F_u^-(u), \quad (4)$$

As a result, we obtain the following function F:

$$F(x,u) = F_0(x) + F_x^+(x)F_u^+(u) + \\ + F_x^-(x)F_u^-(u) \quad (5)$$

The functions of a single variable $F_0(x), F_x^\pm(x), F_u^\pm(u)$ and the function G(x) are called characteristic functions of a resistor.

The characteristic functions $F_x^\pm(x), F_u^\pm(u)$ are not unambiguously defined by equations (4). Let us assume that the functions $F_x^\pm(x)$ are normalized to unity, i.e., they are equal to 1 at characteristic points (at one of the boundary points).

. The characteristic functions $F_0(x), F_x^\pm(x)$ define how quickly a resistor's state changes during programming and erasing, depending on its current state. It is natural to assume that they are continuous on [0,1], positive on (0,1), and satisfy the following relations:

$$F_0(0) = 0, F_x^+(0) = 1, F_x^+(1) = \\ = 0, F_x^-(0) = 0, F_x^-(1) = 1 \quad (6)$$

A typical example of such functions:

$$F_0(x) = f x^{\gamma_0}(1-x)^{\gamma_1}, F_x^+(x) = \\ = (1-x)^{\beta_+}, F_x^-(x) = x^{\beta_-}, \quad (7)$$

where γ_0, β_+, β are positive; γ_1 is non-negative; f is a positive coefficient. Note that for $\gamma_1>0$ we have $F_0(1) = 0$, and for $\gamma_1=0$, $F_0(1) = 1$.

The characteristic functions $F_u^+(u), F_u^-(u)$ define the relationships between the rate of change of the state variable x and the positive or negative voltage applied to the resistor, respectively. The most typical type of functions $F_u^\pm(u)$ is an exponent function defined on the respective semi-axes:

$$F_u^+(u) = A_+\theta(u)u^{\alpha+}, F_u^-(u) = \\ = -A_-\theta(-u)(-u)^{\alpha-} \quad (8)$$

with positive coefficients $A_\pm$ and values $\alpha_\pm$.

Let a periodic piecewise constant voltage $u(t)$ be applied to the variable resistor. Let us denote u_k, k=1,…, K as the voltages, and T_k as the duration of the voltage application during one period T. The change in the resistor's state during one period of the control signal (voltage) according to equation (1) and considering equation (3) can be expressed as

$$x(t+T) - x(t) = F_0(x)T + \\ + \textstyle\sum_{k,u_k>0} F^+(x,u_k)T_k + \sum_{k,u_k<0} F^-(x,u_k)T_k \quad (9)$$

We are interested in slow (averaged over the signal period) changes in the resistor's state (we neglect small fluctuations of the state variable within the period due to their insignificance when a high-frequency signal is applied). For such changes, the equation is

$$\frac{dx}{dt} = \frac{x(t+T)-x(t)}{T} = F_0(x) + \\ + \textstyle\sum_{k,u_k>0} F^+(x,u_k)\tau_k + \sum_{k,u_k<0} F^-(x,u_k)\tau_k \quad (10)$$

where $\tau_k=T_k/T$ is the fraction of time when u_k is nonzero.

Equation (10) was obtained without assuming the factorizability of equation (4). Let us recall that for harmonic signals [7], factorization is required to obtain a correct equation. If we apply factorization to this case, equation (10) can be expressed as

$$\frac{dx}{dt} = F_0(x) + F_x^+(x)M^+ + F_x^-(x)M^-, \quad (11)$$

where $M^+ = \sum_{k,u_k>0} F_u^+(u_k)\tau_k$, $M^- = \sum_{k,u_k<0} F_u^-(u_k)\tau_k$ are the values that represent the responses to positive and negative voltages, respectively.

The second term on the right-hand side of equation (10) or (11) is positive, while the first and third terms are negative. Therefore, the right-hand side of equations (10) and (11) can be either positive or negative. Considering the properties of the characteristic functions expressed in (6), we obtain that when $x = 0$, the right-hand side of equation (11) is positive, and when $x = 1$, it is negative. This means that the equilibrium point equation (11) is

$$P(x) = 0, \quad (12)$$

where $P(x)$ is the right-hand side of equation (11) (or (10) in a more general case), always has a solution within the (0,1) range. For real-life (and not too exotic) characteristic functions, this is the only solution. Let us denote it as x_{st}. According to (11), for $x<x_{st}$, $dx/dt>0$, and for $x>x_{st}$, $dx/dt<0$. Equation (11) describes the monotonic approach of the state variable x to the equilibrium point x_{st}. The approximation rate is P(x). The rate characterizes the efficiency of crossbar programming using a piecewise-constant high-frequency signal.

The equilibrium point equation (12) used in equation (11) can be expressed as

$$F_x^+(x)M^+ = -F_0(x) - F_x^-(x)M^-. \quad (13)$$

Usually, the crossbar programming rate is higher than the rate of spontaneous relaxation. Therefore, the term $F_0(x)$ in the programming equation and equilibrium point equation can be neglected. The resulting equilibrium point equation is

$$\frac{F_x^-(x)}{F_x^+(x)} = \frac{M^+}{-M^-}. \quad (14)$$

Under reasonable assumptions, the left side of equation (14) increases indefinitely starting from 0 as the variable x changes from 0 to 1. The right side is a positive number for a given signal $u(t)$. The value of $\mu = M^+/(-M^-) = M^+/|M-|$ defines the position of the equilibrium point. For $\mu\to 0$ $x_{st}\to 0$, and for $\mu\to\infty$ $x_{st}\to 1$. Usually, $x_{st}(\mu)$ is monotonically increasing.

For characteristic functions (7), equation (14) takes the form

$$\frac{x^{\beta_-}}{(1-x)^{\beta_+}} = \mu. \quad (15)$$

This equation has a unique solution in the (0, 1) range. An explicit expression for x_{st} can only be obtained for certain exponents. For example, for $\beta_+ = \beta_- = \beta$ we obtain

$$x_{st} = \frac{\mu^{1/\beta}}{1+\mu^{1/\beta}}. \quad (16)$$

We also get more cumbersome explicit expressions for x_{st} for $\beta_+=2\beta_-$ and $2\beta_+=\beta$. The $x_{st}(\mu)$ relationships for different values of β_+, β_- are similar. This comes from the fact that when $\mu<<1$ (refer to equation (15)), it follows that $x_{st} \approx \mu^{1/\beta_-}$, and when $\mu>>1$, we have $x_{st} \approx 1-\mu^{-1/\beta_+}$.

To assess the effect of spontaneous relaxation, we substituted expressions (7) into equation (13). The resulting equation is

$$\frac{x^{\beta_-}}{(1-x)^{\beta_+}} = \mu - \frac{f}{-M^-}x^{\gamma_0}(1-x)^{\gamma_1-\beta_+} (17)$$

It differs from equation (15) by an additional negative term on the right-hand side, proportional to $f/|M^-|$. The effect of this term is an effective reduction of μ, i.e., a shift of the equilibrium point to the left (towards the boundary point x=0). In particular, for $\gamma_0 = \beta_+ = \beta_- = \beta, \gamma_1 = 0$, we obtain a simple expression

$$x_{st} = \frac{\hat{\mu}^{1/\beta}}{1+\hat{\mu}^{1/\beta}}, where\ \hat{\mu} = \frac{M^+}{-M^-+f} = \frac{\mu}{1-f/M^-}, \quad (18)$$

similar to expression (16), but with a renormalized μ.

In real-life applications, the crossbar programming rate should be sufficiently high to avoid the effects of spontaneous relaxation of the resistor state during the process. In equations (10) and (11), the first term on the right-hand side can be discarded. The change in the resistor's state is defined by M^+, M^- (or similar values if equation (10) is not factorized). The ratio of these values defines the steady state of the resistor when the control signal is applied.

How do signal parameters affect M^+, M^-, i.e., the programming? The signal parameters are the non-zero voltage levels u_k, k=1,…,K and the relative durations of these voltages τ_k, k=1,…,K. The right-hand sides of equations (10) and (11) vary linearly with τ_k, but their dependence on the voltage levels u_k may be nonlinear. There are usually extensive options for adjusting the programming rate and the position of the equilibrium point. Let us consider these options for the simplest signal with two non-zero voltage levels. It is feasible to use such signals for crossbar programming.

3. The Simplest Control Signal

Let a periodic piecewise constant voltage with two non-zero levels, positive and negative, be applied to the variable resistor. We denote them as u_+ and u_-. Let us denote the respective relative periods of voltage application as τ_+, τ_-. Then equation (10) takes the form

$$\frac{dx}{dt} = F_0(x) + F^+(x,u_+)\tau_+ + F^-(x,u_-)\tau_-. (19)$$

For constant u_+, u_-, there is no need to assume the factorizability of the functions $F^+()$, $F^-()$. It is sufficient to know only their dependence on x for a given value of the second argument. However, if we want to use signals with different amplitudes, we also need to know the dependence of $F^+()$, $F^-()$ on the second argument. Assuming the validity of decompositions (4), we arrive at equation (11), where

$$M^+ = F_u^+(u_+)\tau_+, M^- = F_u^-(u_-)\tau_-, \quad (20)$$

The values M^+, M^- are proportional to τ_+, τ_- respectively, and their dependencies on u_+, u_- are represented by characteristic functions. These dependencies can be obtained by analyzing the simplest signal. For this, we need to fix three of the four signal parameters and change only one: u_+ or u_-. By measuring the programming rate at different values of the variable parameter, we can determine the type of the characteristic function.

By changing u_+, u_- voltages simultaneously (provided that their ratio is constant), we can check whether M^+, M^-vary uniformly as the signal amplitude changes. If the exponential characteristic functions (8) have identical exponents: $\alpha_+=\alpha_-$, μ does not depend on the signal amplitude, and the position of the equilibrium point is virtually independent of the amplitude. For linear characteristic functions $F_u^+(u), F_u^-(u)$ ($\alpha_+=\alpha_-=1$), each of the M^+, M^- values depend linearly only on its combination of $u_+\tau_+$, $u_-\tau_-$. In this case, the number of signal parameters affecting the programming process is reduced to two.

If we want to simplify the control signal even further, additional constraints should be introduced.

For a signal with a zero mean value, the condition is

$$u_+\tau_+ + u_-\tau_- = 0. \quad (21)$$

If such a relationship exists, we have a signal with amplitude u_+ (or $-u_-$) with its shape defined by τ_+, τ_-. The positions of the constant intervals within the signal period are irrelevant. If the characteristic functions $F_u^+(u), F_u^-(u)$ are linear, M^+, M^- depend on a single combination of the $u_+\tau_+$ parameters. This is not so if the $F_u^+(u), F_u^-(u)$ functions are nonlinear on the respective half-axes. We can use this difference to determine whether the characteristic functions are nonlinear.

The additional constraints that simplify the signal are as follows.

$$u_- = -u_+ \quad (22)$$

$$\tau_+ = \tau_- \quad (23)$$

Note that for a signal with a zero mean value, one of the equalities (22) and (23) implies the other. In the next section, the signals that satisfy conditions (22) and (23) are used to program the crossbar. For such a signal, there are only two independent parameters that determine M^+, M^- and, consequently, the crossbar programming rate (the rate of change in the resistor's conductivity). According to (19), (20), in this case

$$P(x, u_0, \tau) = F_0(x) + F^+(x, u_0)\tau + \\ + F^-(x, -u_0)\tau, \quad (24)$$

$$M^+ = F_u^+(u_0)\tau, M^- = F_u^-(-u_0)\tau, \quad (25)$$

$$P(x, u_0, \tau) = F_0(x) + F_x^+(x)F_u^+(u_0)\tau + \\ + F_x^-(x)F_u^-(-u_0)\tau, \quad (26)$$

where $\tau=\tau_+=\tau_-$ is the duty ratio, and $u_0=u_+=-u_-$ is the signal amplitude. Since $P(x, u_0,\tau)$ is the right-hand side of the programming process equation (considering its dependence on the signal amplitude and its duty ratio). Equation (26) applies to the case when functions $F^+(x,u)$, $F^-(x,u)$ are factorized, and equation (24) applies to the general case. There is no fundamental difference between these cases. For certainty, we will use equation (26), and on the right-hand side of the equality, we can discard the first term that represents spontaneous relaxation.

4. Crossbar Array Programming Procedure

The crossbar resistor R_j^i connects the i^{th} word line to the j^{th} bit line. Voltage sources apply potentials to the lines. The voltage applied to the resistor R_j^i is

$$u_j^i = V^i - V_j, \quad (27)$$

where V^i, V_j are the potentials of the i^{th} word line to the j^{th} bit line.

Suppose the following potentials are applied to the lines:

$$V^{i\neq k}(t) = 0, \\ V^k(t) = V_0\sigma(t/T), \quad (28) \\ V_j(t) = V_0\sigma(t/T + \delta_j),$$

where k is the index of the selected row, V_0 is the amplitude of the signals; δ_j is the phase shift ($0\leq\delta_j<1$), $\sigma(y)$ is a periodic piecewise constant function with period 1, for which

$$\sigma(y) = 1, 0 < y < 1/2, \\ \sigma(y) = -1, 1/2 < y < 1 \quad (29)$$

(The values of the function at the points of discontinuity are irrelevant). Function $\sigma(y)$ is a piecewise constant equivalent of the sine function. It can even be defined using the sin() function:

$$\sigma(y) = sign(\sin(2\pi y)) \quad (30)$$

where sign() is the sign function.

The distribution of potential represented by equation (29) means that a standard signal is applied to the k^{th} word line, while the other word lines are grounded. Signals obtained from a standard signal by a phase shift (in the time domain) apply to the bit lines. The phase shift is specific to each bit line. Resistors connected to the k^{th} word line are selected.

The voltage distribution across the crossbar resistors according to (27), (28) is as follows.

$$u_j^{i\neq k}(t) = -V_0\sigma(t/T + \delta_j), \\ u_j^k(t) = V_0(\sigma(t/T) - \sigma(t/T + \delta_j)) \quad (31)$$

The half-selected resistors are affected by the simplest signals that satisfy conditions (22) and (23). The essential parameters of these signals $\tau=1/2$, $u_0=V_0$ are identical for all half-selected resistors.

The voltages applied to the selected resistors are also the simplest signals satisfying conditions (22) and (23). For them, $u_0=2V_0$, but the parameter τ depends on the phase shift: for resistor R_j^k it is equal to

$$\tau_j = min(\delta_j, 1 - \delta_j) \quad (32)$$

When δ_j changes from 0 to 1/2, the parameter τ_j increases linearly from 0 to 1/2, and when δ_j changes from 1/2 to 1, the parameter τ_j decreases linearly from 1/2 to 0. To obtain the full range of the parameter τ variation, it is sufficient to consider half of the phase shift range. Assuming that $0\leq\delta_j\leq1/2$, according to (32) we obtain

$$\tau_j = \delta_j \quad (33)$$

Let us express equations to represent the resistor state changes under voltages (31), assuming that the crossbar resistors are identical, the baseline state of the resistors is x_b, and small deviations from the baseline state x-x_b are used for crossbar programming. Using equation (26), we get

$$\Delta x_j^{i\neq k} = P(x_b, V_0, 1/2)t_r = \\ = (F_x^+(x_b)F_u^+(V_0) + F_x^-(x_b)F_u^-(-V_0))\frac{t_r}{2}, \\ \Delta x_j^k = P(x_b, 2V_0, \delta_j)t_r = \\ = (F_x^+(x_b)F_u^+(2V_0) + F_x^-(x_b)F_u^-(-2V_0))t_r\delta_j \quad (34)$$

(where t_r is the programming period). Half-selected resistors experience an identical change in their states, independent of the phase shifts. Unlike that, selected resistors change their states by an amount proportional to the phase shift, provided that the proportionality coefficient is not zero.

To avoid changes to the states of half-selected resistors, we can select a baseline state identical to the steady state under a standard signal (first equation (32)): $x_b = x_{st}(u_0 = V_0)$. In this case, for half-selected resistors$\Delta x \approx 0$. The steady state under voltage applied to the selected resistors should differ significantly from the baseline state: $x_b \neq x_{st}(u_0 = 2V_0)$. Otherwise, changes in the states of the selected resistors would be insignificant. As we showed in Section 3, such an unfavorable case occurs when the characteristic functions $F_u^+(u), F_u^-(u)$ change similarly along their semi-axes. The values α_+, α_- are equal for exponential characteristic functions. Here, $x_{st}(u_0 = V_0) = x_{st}(u_0 = 2V_0)$ (spontaneous relaxation is discarded), so it is not possible to avoid affecting half-selected resistors by selecting an appropriate baseline state.

When the baseline state is arbitrary, the changes to the states of half-selected resistors must be compensated for. For this, we can apply a constant voltage between all word and bit lines:

$$V^i(t) = V^0, \\ V_j(t) = 0. \quad (35)$$

The voltage V^0 and the period of its application t_{r2} (second programming step) must satisfy the condition:

$$\left(F_x^+(x_b)F_u^+(V^0) + F_x^-(x_b)F_u^-(V^0)\right)t_{r2} = \\ = P(x_b, V_0, 1/2)t_r = \\ = (F_x^+(x_b)F_u^+(V_0) + F_x^-(x_b)F_u^-(-V_0))\frac{t_r}{2}. \quad (36)$$

The polarity of the constant voltage depends on the selected baseline state. When $x_b < x_{st}(u_0 = V_0)$, the voltage V^0 must be negative, and when $x_b > x_{st}(u_0 = V_0)$, it must be positive.

After two programming steps, we have the following changes in the states of the resistor

$$\Delta x_j^{i\neq k} = 0, \\ \Delta x_j^k = (F_x^+(x_b)F_u^+(2V_0) + \\ + F_x^-(x_b)F_u^-(-2V_0))t_r\delta_j - \\ -(F_x^+(x_b)F_u^+(V_0) + F_x^-(x_b)F_u^-(-V_0))\frac{t_r}{2}. \quad (37)$$

As a result, the states of the half-selected resistors remain unchanged, while the changes in the states of the selected resistors linearly depend on the phase shifts. The full range of changes in the state of the selected resistor for a phase shift change from 0 to 1/2, depends on the relative positions of the states x_b, $x_{st}(u_0 = V_0)$, $x_{st}(u_0 = 2V_0)$. In a typical case, $x_b < x_{st}(u_0 = V_0) < x_{st}(u_0 = 2V_0)$, the zero change of the state for the selected resistor is within the available range Δx. Since the programming time t_r is arbitrary, we can use any (small) value Δx.

In this way, arbitrary (within reasonable limits) changes to the states of the resistors in the selected crossbar line can be made. By repeating this procedure for each word line, we can program the crossbar. Both row-wise and column-wise programming processes are possible. Actually, the names like “rows” and “columns” or “horizontal” and “vertical” are only conventional. Rotate the crossbar by 90 degrees to turn rows into columns, and columns into rows.

We can try to change multiple crossbar lines by applying the simplest signal to them (as described in (28)). Such signals may have phase shifts relative to each other. However, in this case, we cannot arbitrarily change the states of the selected resistors because the number of control parameters is insufficient. Only some special cases can be achieved.

5. Conclusion

Arbitrary conductance matrices can be generated by applying simple high-frequency piecewise-constant signals. Programming a pristine crossbar is a multi-step process: at each step, a single row or column is programmed. After initial programming, the crossbar requires only minor adjustments to individual rows or columns.

The convenience and the very possibility of using high-frequency signals for crossbar programming depend on the characteristic functions of the resistors. In particular, the most favorable conditions are when the function $F_I^+(I)$ grows faster than the function $-F_I^-(-I)$ as the current I increases [7]. In this case, the entire range of resistor states is available in principle, and it is easy to obtain both positive and negative Δx values. Otherwise, the available range is smaller, there are fewer baseline state options, and the programming time increases because large currents cannot be used to obtain positive Δx offsets. For the $F_0(x), F_x^+(x), F_x^-(x)$ functions, we assumed that conditions (6) are satisfied. If not, high-frequency signals may not be used. However, conditions (6) seem quite natural and were even tested experimentally [6,10].

The programming procedure and its prerequisites are similar for piecewise-constant and harmonic signals [8]. The question arises: which signal is more convenient in real-life applications? On the one hand, harmonic signals are familiar to radio engineers and compatible with AC circuitry (primarily capacitors), so we can build on the extensive experience in linear RF circuits (for example, we can use various frequency ranges to separate programming and reading). On the other hand, the use of piecewise-constant signals enables more precise and convenient control over the programming process. Unlike a harmonic signal, a piecewise constant signal does not necessarily have a zero mean. In particular, a constant signal is well compatible with a piecewise constant signal, since it is a special case of the latter. For piecewise-constant signals, the result of programming depends linearly on the phase shift (Eq. (37)), in contrast to the nonlinear dependence for harmonic signals [8]. With this, the phase shifts can be found faster and more accurately. Moreover, for advanced digital electronics, the generation of piecewise-constant signals is not too complex.

Piecewise-constant signals are also convenient for analysis and simulation. For given signal values, it is sufficient to know the rate of change of the resistor state only at these voltages; to obtain simple estimates, the factorizability of the right-hand side of the programming equation is not required. There are no problems with modifying the signal waveform, which can be used to identify nonlinear properties of the characteristic functions. The waveform of a harmonic signal is fixed. We can use a sum of harmonic signals [11], but this is a more complicated approach. Note that any signal can be approximated by a piecewise-constant signal. However, this is of no practical use, since with a large number of signal levels, simplicity as its main advantage is lost. For crossbar programming, the simplest piecewise-constant signal consists of two nonzero values applied for equal time intervals. The simplicity of this signal suggests promising practical applications for the proposed crossbar programming method.

This study is a part of the FNEF-2024-0001 Deployment of Trusted AI Systems based on New Mathematical and Algorithmic Approaches and Fast Computing Models Compatible with Domestic Computer Hardware (1023032100070-3-1.2.1) government contract granted to Scientific Research Institute for System Analysis of the National Research Centre Kurchatov Institute, Russian.

References

1. Kotov V.B., Beskhlebnova G.A. Specifics of Crossbar Resistor Arrays. // B. Kryzhanovsky et al. (Eds.). Advances in Neural Computation, Machine Learning, and Cognitive Research VI (NEUROINFORMATICS 2022). Studies in Computational Intelligence. Vol. 1064. Cham: Springer. 2023. PP. 292–304. https://doi.org/10.1007/978-3-031-19032-2_31.

2. Adamatzky A., Chua L. Memristor Networks. Springer International Publishing (2014).

3. Advances in Memristors, Memristive Devices and Systems. / Edited by S. Vaidyanathan and C. Volos. Springer International Publishing AG (2017).

4. Kim S. Ju, Kim S., Jang H.W. Competing memristors for brain-inspired computing. iScience 24, 101889, January 22, 2021.

5. Kotov V.B., Beskhlebnova G.A. Generation of the Conductivity Matrix. // B. Kryzhanovsky et al. (Eds.). Advances in Neural Computation, Machine Learning, and Cognitive Research V (NEUROINFORMATICS 2021). Studies in Computational Intelligence. Vol. 1008. Cham: Springer. 2022. PP. 276-284.

6. Surazhevsky I.A. at all. Noise-assisted persistence and recovery of memory state in a memristive spiking neuromorphic network.Chaos, Solitons and Fractals. 146 (2021). 110890.

7. Beskhlebnova G.A., Kotov V.B. The Variable Resistor Under a High-Frequency Signal. // B. Kryzhanovsky et al. (Eds.). Advances in Neural Computation, Machine Learning, and Cognitive Research VII (NEUROINFORMATICS 2023). Studies in Computational Intelligence. Vol. 1120. Springer Nature Switzerland AG. 2023. PP. 257–266. https://doi.org/10.1007/978-3-031-44865-2_28.

8. Kotov V.B., Beskhlebnova G.A. Use of High-Frequency Signals to Generate a Conductivity Matrix. // B. Kryzhanovsky et al. (Eds.). Advances in Neural Computation, Machine Learning, and Cognitive Research VIII (NEUROINFORMATICS 2024). Studies in Computational Intelligence. Vol. 1179. Springer Nature Switzerland AG. 2025. PP. 265-272.

9. Kotov V.B., Beskhlebnova G.A. Local Point Recording of Information into a Crossbar Resistor Array // SRISA Proceedings. Vol. 14. No. 4 Pp. 33–40 (in Russ.)

10. Kotov V.B., Yudkin F.A. Modeling and Characterization of Resistor Elements for Neuromorphic Systems. Optical Memory and Neural Networks (Information Optics). 2019, v.28, No.4, P. 271-282.

11. V.B. Kotov, Z. B. Sokhova. Two-frequency recording of information into a resistor array. // B. Kryzhanovsky et al. (Eds.). Advances in Neural Computation, Machine Learning, and Cognitive Research IX (NEUROINFORMATICS 2025). Studies in Computational Intelligence. Vol. 1241. Springer Nature Switzerland AG. 2026. PP. 463-476.