Numerical optimization of the transfer function of the intelligent building management system

Pavel Sadchikov*, *Tatyana* Khomenko and *Galina* Ternovaya

Astrakhan State University of Architecture and Civil Engineering, Department of CAD and modelling systems, 414056, Astrakhan, Tatischeva str., Russia

Abstract. The paper deals with structural-parametric models for describing dynamic processes of technical systems of an intelligent building. The task of searching for the transfer function of the synthesized elements and devices of its information-measuring and control systems based on the Mason method is formalized. The components of the transfer function are presented in the form of characteristic polynomials in the structural scheme of the energy-information model of the circuit. The results of a comparative analysis of search methods for multiple real and complex conjugate polynomial roots are presented. To organize their search, an iterative method of unconditional optimization of Fletcher-Reeves was chosen. This conjugate gradient method allows to solve the problem of numerical optimization in a finite number of steps and shows the best convergence in comparison with the methods of the fastest descent, with the same order of difficulty of performing the steps of the algorithm.

1. Introduction

Management of an intelligent building is based on the collection and processing of information coming from elements of controlled engineering systems, represented as an extensive network of sensors and actuators [1-9]. The incoming data characterize the external impact and the response of the chain of a certain physical nature. To be able to assess changes in its parameters in a structurally formalized language for describing processes in technical devices, it is necessary to build an energy-information model of the circuit [10, 11]. According to this model, the relationship between the quantities and parameters of the chains is determined by formula transformations through a system of energy, static and dynamic criteria.

The criteria of the energy-information model of the circuit, provided that the parameters are quasi-constant, can be represented in the operator form:

* Corresponding author: pn_sadchikov@mail.ru

$$\begin{array}{ll} U(p) \approx R(p)\, I(p) & I(p) \approx G(p)\, U(p) \\ I(p) \approx C(p)\, U(p) & U(p) \approx W(p)\, I(p) \\ U(p) \approx L(p)\, I(p) & I(p) \approx D(p)\, U(p) \end{array} \quad (1)$$

where p – impulse, U – impact, I – reaction, R – resistance, G – conductivity, C – capacity, W – rigidity, L – inductance, D – deductive chain.

In case parameters *R, C, L, W, G, D* are constant, then their images in the operator form take the form:

$$\begin{array}{ll} G(p) = G & R(p) = R \\ L(p) = pL & C(p) = pC \\ W(p) = {}^{W}\!/_{p} & D(p) = {}^{D}\!/_{p} \end{array} \quad (2)$$

2. Methods

Since the main task of the system's core is to obtain an analytical function of a parametric block diagram at a given point, a method was chosen based on obtaining the Laplace map of the desired transfer function and its translation into the original area.

In this case, the operation of each individual circuit circuit block can be described as a linear differential equation:

$$a_n Y^{(n)}(t) + a_{n-1} Y^{(n-1)}(t) + \ldots + a_0 = b_m X^{(m)}(t) + b_{m-1} X^{(m-1)}(t) + \ldots + b_0 \quad (3)$$

where *X(t)* – the variation of the input signal;

Y(t) – the variation of the output signal;

$a_n, a_{n-1}, a_0, b_m, b_{m-1}, b_0$ – numerical coefficients.

Thus, a parametrically complex block diagram of an information-measuring and control system of an intelligent building is generally described by a system of equations. So, for example, for the scheme (Fig.):

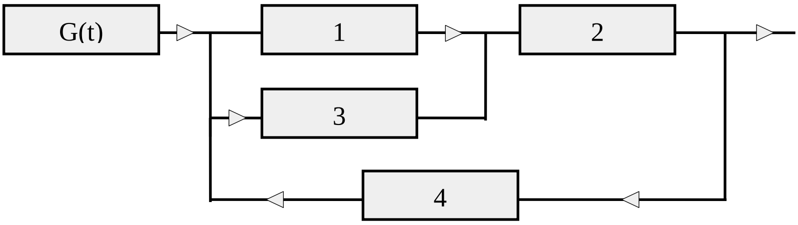

Fig.1. Example of a parametric block diagram

the system of linear differential equations is represented as:

$$\begin{cases} a_{n1} Y_1{}^{(n1)}(t) + a_{n1-1} Y_1{}^{(n1-1)}(t) + \cdots + a_{01} = b_{m1} X_1{}^{(m1)}(t) + b_{m1-1} X_1{}^{(m1-1)}(t) + \cdots + b_{01} \\ a_{n2} Y_2{}^{(n2)}(t) + a_{n2-1} Y_2{}^{(n2-1)}(t) + \cdots + a_{02} = b_{m2} X_2{}^{(m2)}(t) + b_{m2-1} X_2{}^{(m2-1)}(t) + \cdots + b_{02} \\ a_{n3} Y_3{}^{(n3)}(t) + a_{n3-1} Y_3{}^{(n3-1)}(t) + \cdots + a_{03} = b_{m3} X_3{}^{(m3)}(t) + b_{m3-1} X_3{}^{(m3-1)}(t) + \cdots + b_{03} \\ a_{n4} Y_4{}^{(n4)}(t) + a_{n4-1} Y_4{}^{(n4-1)}(t) + \cdots + a_{04} = b_{m4} X_4{}^{(m4)}(t) + b_{m4-1} X_1{}^{(m4-1)}(t) + \cdots + b_{04} \\ X_1(t) + X_3(t) = Y_G(t) + Y_4(t) \\ X_2(t) = Y_1(t) + Y_3(t) \\ X_4(t) = Y_2(t) \end{cases}$$

(4)

The search for a solution to the resulting system of equations reduces to the definition of an analytic representation of functions $X_1(t), \dots, X_n(t), Y_1(t), \dots, Y_m(t)$ for a given $Y_G(t)$, produced by the generator *G(t).*

To determine the dynamic characteristics of systems built according to similar rules, we use the Mason automated calculation method. [12, 13]. The main idea of this method is that the system is represented as a directed graph, in which arcs are equivalent to elementary links (blocks), and vertices - to signal transmission lines. The direction of the arc corresponds to the direction of movement of the signal. The control point at which the desired function will be calculated can only be a vertex. Since each vertex may be incident, both along the incoming and outgoing arcs, the signal at the vertex is a superposition of the output signals of the blocks corresponding to the arcs entering the vertex. The generator is represented by an arc, the initial vertex of which can only be outgoing.

The Laplace transfer function in a fixed r-vertex of a directed graph is calculated by the formula:

$$L\big(f_r(t)\big) = \sum_{i=1}^{q} L\big(g_i(t)\big)\, W_{xi,r} \tag{5}$$

where $L\big(f_r(t)\big)$ – is the mapping of the desired process;

q – number of generators;

$L\big(g_i(t)\big)$ – display of the function generated by the i-th generator;

xi – the final vertex of the arc corresponding to the i-th generator;

$W_{xi,r}$ – is the transfer function of the circuit that converts the signal coming from the vertex xi into the signal arriving at the vertex r.

As a result of the implementation of the Mason algorithm, we obtain the transfer function of the entire system:

$$W_{xy}(t) = \frac{\sum_{i=1}^{m}[W_{xy}^{i}(t)\ \Delta_i]}{\Delta} \tag{6}$$

where $W_{xy}^{i}(t)$ – the transfer function i of a simple path from the vertex y to the vertex x, equal to the product of the transfer functions of the arcs included in this path;

m – total number of such paths;

Δ – the determinant of the graph, calculated by the formula

$$\Delta = 1 - \sum_j W_{0j} + \sum_{j,k} W_{0j} W_{0k} - \sum_{j,k,1} W_{0j} W_{0k} W_{01} + \dots \tag{7}$$

The inverse Laplace transform allows you to get the original function *f (t)* in its image *F(p)*:

$$f(t) = \mathcal{L}^{-1}\{F(p)\} = \frac{1}{2\pi i \int_{c-j\infty}^{c+j\infty} F(p) e^{pt} dp} \tag{8}$$

3. Results and Discussion

The use of this expression is very difficult, especially for complex expressions *F (p).* Therefore, more acceptable methods of determining the original *f (t)* were developed. To move from the image of *L {F (p)}* to the original, you need to convert it to a rational fraction, decompose it into simple fractions and, using table transformations, present the sum of the simplest fractions as the sum of their corresponding originals.

As a result of the implementation of the developed algorithms for working with an oriented graph, the transfer function of the structural scheme, defined by the Mason formula, can be expressed in terms of the transfer functions of the elementary links.

Consider the transformation of the transfer function of the structural scheme from the display to the original, i.e. to real dynamic response:

$$W(p) = \frac{Y(p)}{X(p)} = \frac{b_m p^m + b_{m-1} p^{m-1} + \dots + b_0}{a_n p^n + a_{n-1} p^{n-1} + \dots + a_0} \tag{9}$$

Each rational function of p appearing in the denominator of the transfer function of the structural scheme is represented as a polynomial and is a characteristic polynomial of this structural scheme, and an equation of the form:

$$P_n(p) = a_0 + a_1 p + a_2 p^2 + a_3 p^3 + \dots + a_n p^n = 0 \qquad (a_n \neq 0) \tag{10}$$

is a characteristic equation of the structural scheme, which has n roots, among which can be both real multiples and complex conjugates.

It is known that any regular rational fraction $\frac{Y(p)}{X(p)}$, whose denominator is factorialized:

$$X(p) = (p - p_1)^{n1}\left((p - p_2)\right)^{n2}(p^2 + r_1 p + q_1)^{s1} \dots (p^2 + r_m p + q_m)^{sm}, \tag{11}$$

can be represented (and in this unique way) in the form of the following sum of simple fractions:

$$\frac{Y(p)}{X(p)} = \frac{b_1}{p-p_1} + \frac{b_2}{(p-p_1)^2} + \dots + \frac{b_{k1}}{(p-p_1)^{k1}} + \dots + \frac{b_{km_k}}{(p-p_1)^{m_k}} + \dots +$$

$$+ \frac{c_1 p + d_1}{p^2 + r_1 p + q_1} + \frac{c_2 p + d_2}{(p^2 + r_1 p + q_1)^2} + \dots + \frac{c_{s1} p + d_{s1}}{(p^2 + r_1 p + q_1)^{s1}} + \dots + \frac{c_{sm_s} p + d_{sm_s}}{(p^2 + r_s p + q_s)^{m_s}}, \tag{12}$$

each of which is a transfer function of the link of the structural diagram.

For such a representation $\frac{Y(p)}{X(p)}$ you need to find the roots of the polynomial *X (p)* and determine the coefficients.

Case 1. The polynomial *X (p)* of the denominator has real roots $p_k$of multiplicity m_k.

1) the degree of the numerator m of the polynomial *Y (p)* is greater than the degree of the denominator *n* of the polynomial *X (p):*

when *n≤m*, the numerator is divided by the denominator according to Gorner's scheme, then step 2),

2) the degree of the numerator m of the polynomial *Y (p)* is less than the denominator n of the polynomial *X (p):*

for *n≥m*, the transfer functions of the structural scheme are represented as the sum of the elementary fractions corresponding to the roots p_k (of multiplicity m_k) of the polynomial *X (p)*:

$$\frac{Y(p)}{X(p)} = \sum_k \sum_{j=1}^{m_k} \frac{b_{kj}}{(p-p_k)^j} = \sum_k \left[\frac{b_{k1}}{(p-p_k)} + \frac{b_{k2}}{(p-p_k)^2} + \dots + \frac{b_{km_k}}{(p-p_k)^{m_k}}\right] \tag{13}$$

To find the roots of the characteristic equation of the structural scheme, classical methods are used. [14, 15].

Comparative results of methods for finding real multiple roots of a polynomial are presented in table 1.

Table 1. Comparative results of polynomial root finding methods

№	*Method Name* *characteristics*	linear interpolation method (chord method)	Newton's method (the method of tangents)	Stephenson's method
1	*reliability*	high	high	high
2	*rate of convergence*	slow	high	high
3	*application at multiple roots*	-	+	+
4	*the need to calculate derivatives at each step*	-	+	-
5	*number of iterations*	large	small	large
6	*implementation difficulty at program level*	-	+	-

To organize the search for real roots p_k of the multiplicity m_k of the characteristic equation of the structural scheme, the Stephenson method was chosen, despite the fact that an increased number of iterations due to the replacement of approximation by a derivative with a computational expression leads to a new iterative algorithm. However, due to this, there is no need to calculate the derivatives at each step, and the difficulty of implementation at the program level is sharply reduced.

The main iterative steps of the Stephenson method [16, 17] can be represented as follows.

1. Let be x_n – the current approach to the root x_n and $\Delta x = x_r - x_n$.
2. Find the point $x_r^l = x_{n+1}$ as the following x_n root refinement.
3. Then we come to the iterative algorithm:

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n+f(x_n))-f(x_n)} \cdot f(x_n), \tag{14}$$

which underlies the Stephenson method.

For the decomposition of a rational fraction $\frac{Y(p)}{X(p)}$ with the denominator represented as factors, according to the found roots of the polynomial X (p), into simple fractions, it is necessary to calculate the coefficients b_{kj}. It is proposed to use one of the following methods to calculate the b_{kj} coefficients or a combination of these methods [18]:

1. If $m_k = 1$ (p_k – simple root), that $b_{kj} = g(p_k)/f'(p_k)$.
2. Multiply both sides of equality (13) by *f (p)* and equate the coefficients with equal powers of *p* in both sides of the obtained equality.
3. Multiply both sides of the equality (13) on $f(p)$ and successively differentiate the resulting equality. Let be $\varphi_k(p) = f(p)/(p-p_k)^{m_k}$. Then $b_{km_k}, b_{km_{k-1}}, \dots$ sequentially found from the following relationships:

$$\begin{aligned} &g(p_k) = b_{km_k}\varphi_k(p_k) \\ &g'(p_k) = b_{km_k}\varphi_k'(p_k) + b_{km_{k-1}}\varphi_k(p_k) \\ &g''(p_k) = b_{km_k}\varphi_k''(p_k) + 2b_{km_{k-1}}\varphi_k'(p_k) + 2b_{km_{k-2}}\varphi_k(p_k) \\ &g^{(m_k-1)}(p_k) \\ &\qquad = b_{km_k}\varphi_k^{(m_k-1)}(p_k) + m_k b_{km_{k-1}}\varphi_k^{(m_k-2)}(p_k) \\ &\qquad + m_k m_{k-1} b_{km_{k-2}}\varphi_k^{(m_k-3)}(p_k) + \cdots + m_k!\, b_{k1}\varphi_k(p_k) \end{aligned} \tag{15}$$

Case 2. The polynomial $X(p)$of the denominator has complex conjugate roots $p_k$of multiplicity m_k:

1) the degree of the numerator m of the polynomial *Y (p)* is greater than the degree of the denominator n of the polynomial *X (p):*

when n≤m, the numerator is divided by the denominator according to Gorner's scheme, then the item is executed 2),

2) the degree of the numerator *m* of the polynomial *Y (p)* is less than the denominator *n* of the polynomial *X (p):*

for *n ≥ m,* the transfer functions of the structural scheme are represented as a sum of unit fractions corresponding to the roots p_s (of multiplicity m_s) of the polynomial *X (p):*

$$W(p) = c_{s1}\frac{p+d_{s1}}{[(p-a_s)^2+\omega_s^2]} + c_{s2}\frac{p+d_{s2}}{[(p-a_s)^2+\omega_s^2]^2} + \cdots + c_{sm_s}\frac{p+d_{sm_s}}{[(p-a_s)^2+\omega_s^2]^{m_s}} \tag{16}$$

In this case, the elementary fractions correspond to an arbitrary pair of complex conjugate roots : $a_s + j\omega_s$; $a_k - j\omega_k$ multiplicities m_s.

With all the variety and variability of the numerical implementation of algorithms for the organization of their search, one of the methods of conjugate gradients was chosen, namely the iterative method of unconditional Fletcher-Reeves optimization [19].

The methods of conjugate gradients are based on the calculation of the values of only the first derivatives and have positive properties in comparison with the Newtonian methods (Cauchy, Newton and others), which are referred to the methods using second order derivatives. Newton's methods guarantee obtaining a minimum of nonlinear functions of the goal, but it is impossible to judge in advance the number of necessary iterations even for the case of quadratic functions of the goal. The methods of conjugate gradients guarantee the achievement of a minimum of a quadratic goal function in a finite number of steps, not exceeding the value of *n*, the dimension of the space R^n [20]. Such methods are called quadratically convergent, since the rate of convergence is quadratic. This is a significant advantage when compared with conventional gradient methods (the steepest descent method or the coordinate descent method). Conjugate gradient methods converge 4-5 times faster than the fastest descent method. These methods are highly reliable and quickly converge in the vicinity of the minimum point. They allow you to guarantee convergence in a finite number of steps, and the required accuracy can be achieved much earlier.

Comparative results of methods for finding complex conjugate roots of a polynomial are presented in Table 2.

Table 2. Comparative results of methods for finding the roots of a polynomial

№	*Method Title* *characteristics*	gradient method	Newton method	Fletcher-Reeves method
1	*reliability*	high	high	high
2	*rate of convergence*	slow	high	high
3	*the need to calculate the values of the first derivatives*	+	+	+
4	*the necessity of calculating the values of the second derivatives*	-	+	-
5	*implementation difficulty at program level*	-	+	-
6	*computer memory efficiency*	+	-	+

The advantage of the Fletcher-Reeves algorithm is that it does not require time-consuming computations of second partial derivatives like Newton's methods and saves computer memory, since it does not need Hessian matrices used in Newton's method and other second-order methods. At the same time, this method is almost as effective as quasi-Newton algorithms [15].

The Fletcher-Reeves method allows us to find the minimum of a nonlinear objective function of several variables [17]. For this, the complex conjugate roots are represented as a function of two variables:

$$f(x) = f_1(x_1, x_2) + if_2(x_1, x_2), \tag{17}$$

where $x_1 = Re\, x, x_2 = \mathrm{Im}\, x, f_1 = Re\, f, f_2 = \mathrm{Im}\, f$.

Then the problem of finding the roots of a polynomial is equivalent to the problem:

$$F(x) = f_1^2(x_1, x_2) + f_2^2(x_1, x_2) \rightarrow min \tag{18}$$

Such a formulation of the problem extends to the search for all roots of a polynomial, both real multiple and complex conjugate.

Consider the Fletcher-Reeves algorithm in more detail. In this method, the direction of steepest descent is rejected by adding to it the direction used in the previous step. As a result, a sequence of search directions is built. $S^{(k)}$, which are linear combinations of the gradient of the current direction of the fastest descent $\left[-\nabla f(p^{(k)})\right]$, and previous research directions - $S^{(i)}$, i. e.

$$S^{(k+1)} = -\nabla f(p^{(k)}) + \sum_{i=1}^{k} \alpha_i S^{(i)}. \tag{19}$$

The values of weighting coefficients α_i are chosen so that the new direction $S^{(k)}$ it was associated with all previous directions. With this criterion for the end of the search is the condition:

$$\left(\nabla^2 f(p^{(k)}), S^{(k)}\right) = 0 \tag{20}$$

It is proved that: $\alpha_1 = \alpha_2 = \cdots = \alpha_{k-1} = 0, \quad \alpha_k = \frac{\|\nabla f(p^{(k)})\|^2}{\|\nabla f(p^{(k-1)})\|^2}$,

where $\|\nabla f(p^{(k)})\|^2$ - the gradient modulus determines the rate of increase or decrease of the function in the direction of the gradient or anti-gradient, respectively. This condition allows us to build a fast and efficient algorithm for searching for extrema.

As a result, the Fletcher-Reeves algorithm can be written as follows:

1) Set $\varepsilon > 0,\ p^1$ to the starting point, $S^{(1)} = -\nabla f(p^1)$
and set $y^{(1)} = p^{(1)},\ k = j = 1$.

2) If $\|\nabla f(p^{(k)})\| < \varepsilon$, then the end and the optimum will be: $p_* = y^{(j)}$, otherwise the one-dimensional optimization $\min_\lambda f(y^j + \lambda S^j) \to \lambda_j$,
put $y^{(j+1)} = y^{(j)} + \lambda_j S^{(j)}$

3) If $j < n$, go to step 4, otherwise to step 5.

4) Put $S^{(j+1)} = -\nabla f\left(y^{(j+1)}\right) + \frac{\|\nabla f(y^{(j+1)})\|^2}{\|\nabla f(y^{(j)})\|^2} S^{(j)},\ j = j + 1$, go to step 2.

5) Put $p^{(k+1)} = y^{(n+1)},\ y^{(1)} = y^{(n+1)}, S^{(1)} = -\nabla f(y^{(1)}),\ k = k + 1,\ j = 1,$ go to step 2.

4. Conclusion

The Fletcher-Reeves algorithm is sensitive to the accuracy of a one-dimensional search, so when using it, you need to eliminate any rounding errors that may occur. There is no guarantee of convergence always and everywhere in the algorithm, but as shown by practical calculations, the algorithm almost always leads to an approximate optimum. The main advantage of the conjugate gradient method is that it allows to solve the optimization problem in a finite number of steps and shows better convergence in comparison with the methods of the fastest descent, with the same order of complexity of the steps of the algorithm.

Thus, to go from the image to the original of the transfer function of the parametric structural scheme, you must perform the following steps:

1. Find the roots of the denominator of the output transfer function using the Fletcher-Reeves algorithm.
2. Decompose the output transfer function into elementary terms.
3. Translate elementary terms into the original area and present the original output transfer function as a whole.

References

1. V. Zaripova, I. Petrova. *Proceedings of the 23rd International Conference on Systems Engineering, 2014, Las Vegas, NV, Series: Advances in Intelligent Systems and Computing*, **1089**, 365–373 (2015)
2. D. Anufriev, I. Petrova, O. Shikulskaya. *Creativity in Intelligent Technologies and Data Science. CIT&DS 2017. Communications in Computer and Information Science,* 754 (2017)
3. D. Anufriev. *Proceedings of the II International Scientific and Practical Conference*, 58–73 (2010)
4. G. Gemot, D. Stark. *Restructuring Networks: Legacies, Linkages, and Localities in Postsocialism,* 1-32 (1997)
5. T. Zolina, P. Sadchikov. *Industrial and civil construction*, **8**, 14-16 (2012)
6. T. Zolina, P. Sadchikov. *Applied Mechanics and Materials*, **752-753**, 1218-1223 (2015)
7. L. Boronina, P. Sadchikov, S. Tazhieva, E. Moskvicheva. *Water Resources,* **43-4**, 657-662 (2016)
8. N. Braila, K. Khazieva, A. Staritcyna. *Magazine of Civil Engineering*, **6**, 70–77 (2017)
9. T. Zolina, P. Sadchikov. *Industrial and Civil Engineering*, **8**, 19-20 (2012)
10. V. Zaripova, I. Petrova, K. Shumak, Yu. Lezhnina. *Bulletin of MGSU,* **12 (111)**, 1424-1434 (2017)
11. E. Krinitsky. *Magazine of Civil Engineering*, **2**, 16-18 (2010)
12. H. Damerdji, M. Nakayama. *Assoc. Comput. Mach. Trans. Modeling and Comput. Simul.*, **9**, 1-30 (1999)
13. E. Yucesan, L. Schruben. *Assoc. Comput. Mach. Trans. Modeling and Comput. Simul.,* **2**, 82-103 (1992)
14. A. Avramidis, J. Wilson. *Operations Res.*, **44**, 327-346 (1996)
15. F. Vasiliev. *Optimization methods*, 292-300 (2002)
16. N. Bakhvalov, N. Zhidkov, G. Kobelkov. *Numerical methods*, 600 (2003)
17. V. Zaitsev, V. Treshchev. *Numerical methods. Nonlinear equations and optimization*, 86 (2005)
18. G. Korn, T. Korn. *A handbook on mathematics for scientists and engineers*, 45-46 (1970)
19. T. Shup. *The solution of engineering problems on a computer: A practical guide*, 238 (1982)
20. V. Karmanov. *Mathematical Programming*, 186-190 (2004)